NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INTEGRATION OF MULTIPLE UNMANNED SYSTEMS IN AN URBAN SEARCH AND RESCUE ENVIRONMENT

by

Boon Heng Chua

March 2013

Thesis Advisor:	Oleg Yakimenko
Second Reader:	Mark Stevens

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704–0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2013	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE INTEGRATION OF MULTIPLE UNMANNED SYSTEMS IN AN URBAN SEARCH AND RESCUE ENVIRONMENT			**5. FUNDING NUMBERS** N/A
6. AUTHOR(S) Boon Heng Chua			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943–5000			**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER** N/A
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number N/A.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

In view of the local, regional and global security trends over the past decade, the threats of disaster to the populace inhabiting urbanized areas are real and there is a need for increased vigilance. There can be multiple causes for urban disaster: natural disasters, terrorist attack and urban warfare are all viable. This thesis focused on the event in which an urban search and rescue operation is required due to the aftermath of a terrorist activity. Systems engineering techniques were utilized to analyze the problem space and suggested a plausible solution. Application of unmanned vehicles in the scenario enhanced the reconnaissance, intelligence and surveillance capabilities of the responding forces, while limiting the exposure risk of personnel.

One of the many challenges facing unmanned systems in a cluttered environment is a capability to rapidly generate reactive obstacle avoidance trajectories. A direct method of calculus of variations was applied for the unmanned platforms to achieve mission objectives collaboratively, and perform real-time trajectory optimization for a collision-free flight. Dynamic models were created to enable simulated operations within the thesis design scenario. Experiments conducted in an indoor lab verified the unmanned systems' ability to avoid obstacles and carry out collaborative missions successfully.

14. SUBJECT TERMS Quadrotor, UAV, Direct method, Urban environment, Urban Search and Rescue, Trajectory Optimization, Inverse Dynamics, IDVD			**15. NUMBER OF PAGES** 137
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01–280–5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

INTEGRATION OF MULTIPLE UNMANNED SYSTEMS IN AN URBAN SEARCH AND RESCUE ENVIRONMENT

Boon Heng Chua
Civilian, Defence Science and Technology Agency, Singapore
B.Eng, (Mechanical), Nanyang Technological University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2013

Author: Boon Heng Chua

Approved by: Oleg Yakimenko
Thesis Advisor

Mark Stevens
Second Reader

Clifford Whitcomb
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In view of the local, regional and global security trends over the past decade, the threats of disaster to the populace inhabiting urbanized areas are real and there is a need for increased vigilance. There can be multiple causes for urban disaster: natural disasters, terrorist attack and urban warfare are all viable. This thesis focused on the event in which an urban search and rescue operation is required due to the aftermath of a terrorist activity. Systems engineering techniques were utilized to analyze the problem space and suggested a plausible solution. Application of unmanned vehicles in the scenario enhanced the reconnaissance, intelligence and surveillance capabilities of the responding forces, while limiting the exposure risk of personnel.

One of the many challenges facing unmanned systems in a cluttered environment is a capability to rapidly generate reactive obstacle avoidance trajectories. A direct method of calculus of variations was applied for the unmanned platforms to achieve mission objectives collaboratively, and perform real-time trajectory optimization for a collision-free flight. Dynamic models were created to enable simulated operations within the thesis design scenario. Experiments conducted in an indoor lab verified the unmanned systems' ability to avoid obstacles and carry out collaborative missions successfully.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ASEIL	Autonomous Systems Engineering Integration Laboratory
CCS	Central Control Station
DAC	Data Acquisition Card
EMMI	Energy, Matter, Material wealth and Information
FLIR	Forward Looking Infrared
GCS	Ground Control Station
GPS	Global Positioning System
HAZMAT	Hazardous Materials
IDVD	Inverse Dynamics in Virtual Domain
IED	Improvised Explosive Device
IR	Infrared
LCD	Liquid Crystal Display
LQR	Linear quadratic regulator
QuaRC	Quanser Real-time Control
ROV	Remotely Operated underwater Vehicle
SLAM	Simultaneous Locating and Mapping
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
USAR	Urban Search and Rescue
USV	Unmanned Surface Vehicle
UUV	Unmanned Underwater Vehicle
VGTV	Variable Geometry Tracked Vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Considering the local, regional and global security trends over the past decade, the threats of disaster to the populace inhabiting urbanized areas are real and, there is a need for increased vigilance. A background research has indicated that there can be multiple causes for urban disaster: natural disasters, terrorist attack and urban warfare are all viable. A thorough literature review was done in many areas. These areas include: the expected urban environment; the threats to the urban environments; urban search and rescue operations; unmanned systems that were utilized in urban search and rescue (USAR) operations; challenges of using unmanned systems; and finally, the key researches in the area of trajectory optimization, in particular the Inverse Dynamics in Virtual Domain (IDVD) methodology. In essence, the intricacies of operating multiple unmanned systems in a cluttered urban search and rescue environment were explored and discussed. One crucial finding is that in order to maximize the benefits of these unmanned systems, more autonomy during such operations is pivotal. In a resource constrained environment, autonomy of such unmanned systems will free up valuable assets for other critical functions.

Systems engineering techniques were utilized to analyze the problem space and suggested a plausible solution. First the problem, based on the literature review performed, was defined. The boundaries (physical, functional and behavioral) of the system are outlined using the input-output model and the contextual diagram. The stakeholder analysis performed was vital in surfacing the effective needs and capability requirements of the system. Subsequently, the limitations and constrains of unmanned systems were investigated and important considerations were highlighted for the creation of a valid concept of operations. Application of unmanned vehicles in the scenario enhanced the reconnaissance, intelligence and surveillance capabilities of the responding forces, while limiting the exposure risk of personnel. A thesis design scenario was then created to be verified and validated later by experimentation.

One of the many challenges facing unmanned systems in a cluttered environment is a capability to rapidly generate reactive obstacle avoidance trajectories. A direct

method of calculus of variables was applied for the unmanned platforms to achieve mission objectives collaboratively and perform real-time trajectory optimization for obstacles avoidance. The IDVD methodology applied was able to rapidly generate quasi-optimal trajectories that meet the immediate needs of the platform. Dynamic models were created using Simulink to enable simulated operations within the thesis design scenario. With the use of Simulink models, the designs of the desired trajectory are flexible and it offers a great learning platform for apprentices trying to learn and appreciate IDVD.

The laboratory was set up to perform the required experiments to validate and verify the feasibility of the optimization methodology. Capabilities and the limitations of the systems used to perform the experiments were identified for the investigation of two sub-scenes from the thesis design scenario. In scene 1, a single unmanned system scanning the debris horizon for surviving casualties encounters an obstacle that prevents it from proceeding forward. This requires the UAV to make flight adjustments to negotiate over and above the obstacle. In scene 2, multiple UAVs have been deployed in the field to perform auxiliary duties. In particular, 2 UAVs while maneuvering in the cluttered environment are headed toward each other in a collision course. Besides avoiding each other, the UAVs must simultaneously avoid a raised barrier that is blocking their advancements in their respective trajectory. Trajectories in each scene were generated using the methodology described in Chapter IV and executed by the quadrotor platform. The experiments that were conducted verified the cogency of the generated trajectories and validated the unmanned systems' ability to avoid obstacles and carry out collaborative missions successfully.

ACKNOWLEDGMENTS

I would like to thank Professor Oleg Yakimenko for sharing his expertise and valuable advice throughout the thesis journey. He was unfailingly supportive and always approachable.

In addition, I would like to thank Professor Mark Stevens for his advice and continuous motivation throughout the entire thesis process.

Appreciation also goes out to Juan Gonzales and Tommy Pierce, laboratory technicians at the Naval Postgraduate School, for their technical support in and around the Autonomous Systems Engineering and Integration Laboratory; Cameron Fulford, Engineering Manager from Quanser Academic, for providing expert technical assistance on the Quanser Qball-X4 quadrotor hardware.

THIS PAGE INTENTIONALLY LEFT BLANK

I. BACKGROUND AND LITERATURE REVIEW

A. BACKGROUND

With swift advancement in information technology and rapid increase in computing power, the unmanned vehicle arena has seen exponential growth in recent years. Continual growth of the unmanned platforms seen today can be reasonably expected (Castelli 2012). Unmanned vehicles are powered platform that carries no operator and can be either remotely or autonomously controlled. Unmanned vehicles can be organized into five major categories according to the medium through which they travel. These categories are: unmanned air vehicles (UAV), unmanned ground vehicle (UGV), unmanned surface vehicles (USV), unmanned underwater vehicles (UUV) and unmanned spacecraft.

These vehicles are considered to be versatile and can be used to capably complete a wide variety of tasks. In the unmanned vehicle field, the use of unmanned vehicles is dominated by military applications. Besides military applications, non-military use for the unmanned system is on the rise. For instance, the U.S. Department of Energy deploys UGVs to safeguard the Nevada National Security Site; farmers utilize self-driving tractors to fertilize and harvest corps; and for decades autonomous probes have been used by scientists to map the ocean floor (Trivedi 2011). Similar to the robotic devices in automation, unmanned systems are capable of taking on tasks that are deemed “dull, dirty or dangerous” for humans. Some examples of “dull, dirty or dangerous” tasks include intelligence gathering, reconnaissance, hazardous level monitoring and terrain mapping. These tasks require long hours of operations and may involve treading into hostile environment. Having unmanned vehicle capabilities would allow its users the ability to complete missions in a safer and more efficient manner.

As rapid population growth and urbanization continues to change the face of the world, the likelihood of having to deal with an urban emergency increases. According to The World Bank (Figure 1), more than 50% of the World’s populations are living in urban environments (The World Bank 2013).

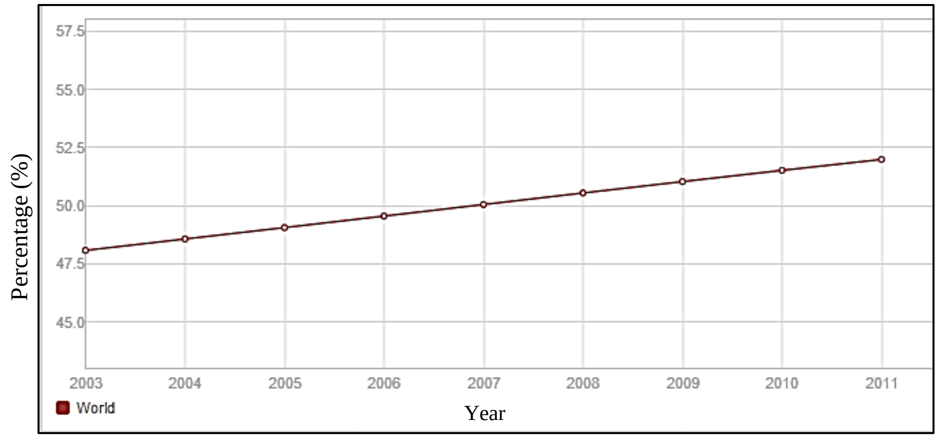


Figure 1. Urban population (% of total population) (From The World Bank 2013).

The statistic is likely to increase yearly as developing countries become more urbanized. The world's urban population is increasing four times as fast as its rural population, and it is undergoing the largest wave of urban growth in history. By 2030, five billion people are projected to be living in urban areas, and 90 percent of the growth will be in the developing world (UNFPA 2012). Figure 2 depicts the countries and territories with 2050 urban populations exceeding 100,000. The circles are scaled in proportion to the projected urban population size (UNICEF 2012).

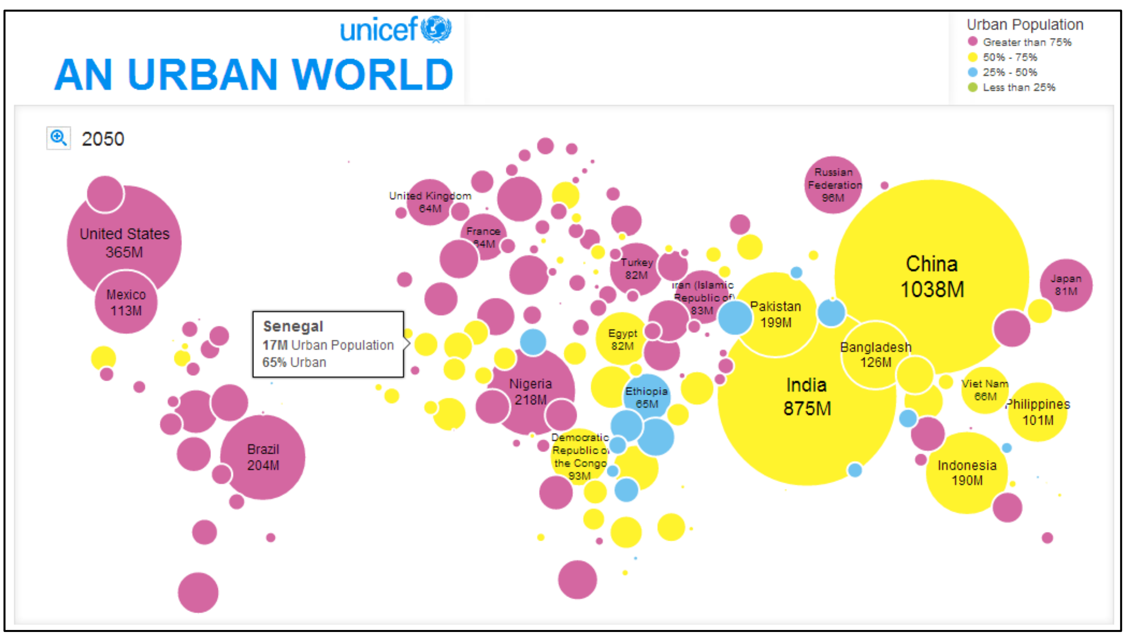


Figure 2. Projected urban population in year 2050 (From UNICEF 2012).

Taking into consideration the local, regional and global security trends over the past decade, the threats of disaster to the populace inhabiting urbanized areas are real and, there is dire need for increased vigilance. There can be multiple causes for urban disaster: natural disasters, terrorist attack or even urban warfare are all viable. The application of unmanned vehicles in these scenarios would possibly enhance the reconnaissance, intelligence and surveillance capabilities of the responding forces. At the same time, human operators are protected from the perils of entering the engagement zone. Disaster response is always a race against time, to get to as many potential survivors in the shortest time possible. However, this primary consideration of moving fast contradicts with the need to also move slowly enough so that one does not disturbs the unstable nature of the ground, and avoid causing further damage to rescue resources and the victims.

This thesis presents the integration of multiple unmanned vehicles in an urban search and rescue environment. The next section details the literature review of previous works performed by scholars and researchers in the area of urban environment, the threats to an urban environment, commercial unmanned vehicles available, and the IDVD control methodology.

B. LITERATURE REVIEW

1. Urban Environment

The urban environment can be described by two components, the physical terrain and the human environment (Rice 2003). Perhaps the greatest reason urban operations are regarded as difficult is that cities present the most complex and dynamic environment imaginable for operations. Understanding the foundation of this environmental complexity is essential to understanding how the other factors are magnified.

Dr. Russell Glenn analyzes the various peculiarities of the urban environment and the impact they have on operations. He provided a number of combat examples to show that understanding how best to allocate resources using such concepts as critical points and density. His work includes examination of environmental differences and the effects they have on time, command and control, and information systems (Glenn 2002).

Dr. Glenn also calls for development of weapons, communications, and support systems to account for the environmental challenges presented by urban areas and the need for joint urban operations doctrine. He asserts that the key to urban operations lies in overcoming the challenges presented by environmental densities which impact operators (Glenn 2002).

FM 3–06 is the U.S. Army's tactical-level urban operations doctrine and it provides the analytical tools for evaluating an urban operation to determine if the operation is necessary for overall mission success (Headquarters, Department of the Army 2006). Within the doctrine, it provides its users with toolset and information to help users determine and manage the impacts that the urban environment has on military operations. Of interest is the highly detailed guidance on analysis of the physical characteristics of the urban environment. In Figure 3, the manual dissects the area of operation into four major dimensions, namely the Airspace, Surface, Super-surface and Sub-surface. The manual also discusses the effects of urban environment on operations.

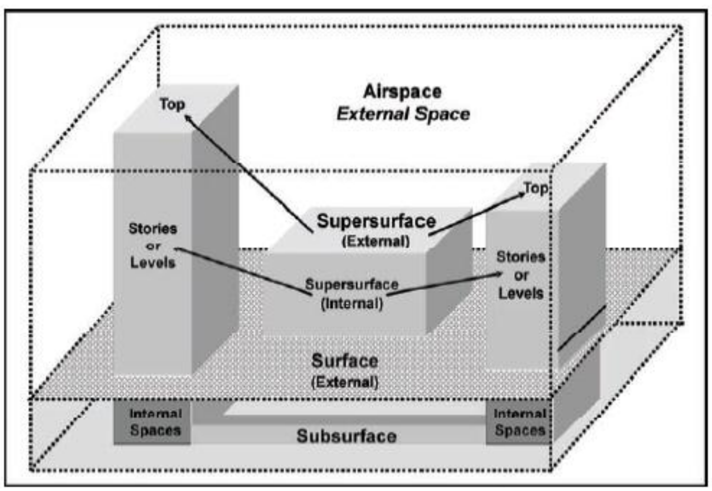


Figure 3. The multidimensional urban environment
(From Headquarters, Department of the Army 2006).

In an urban environment, an unmanned/robotic system will need to traverse seamlessly between both environments while maintaining accurate localization and effective mapping. One of the most glaring limitations of operating traditional unmanned

systems in an urban environment is the limited line of sight with the satellite and the ground control station (GCS) (Nguyen et al. 2009). The termed “urban canyons” are used to describe urban environment where GPS signals are lost or unreliable. It is beneficial to take advantage of multiple localization techniques, including Kalman Filter-based inertial navigation and the laser-based simultaneous localization and mapping (SLAM). Optimization techniques are used to combine the outputs of these techniques to maximize the accuracy of localization.

2. Threats

Threats to the urban environment can arise from quite a few scenarios, most notably due to terrorist activities. Following the September 11 attacks in the U.S., the 2004 Madrid and 2005 London train bombings (U.K. House of Commons 2006) demonstrate the technical sophistication of terrorists and their shift toward a campaign targeting urban environment that can result in mass casualties. Manmade disasters are geographically concentrated with the most vital aspects of the disaster invisible beneath the rubble. Historically, there are some notable and successful attempts to cause serious public harm and unrest by terrorist groups. Some of the major incidents include:

a. Tokyo, Japan: Subway Attack (1995)

On March 20th, 1995, an attack executed by the Japanese cult Aum Shinrikyo was the deadliest and most notorious attack that involved the use of a chemical warfare agent on a target population in a public place. Multiple devices distributed and released in an underground subway station released a total of 600 grams of 2-(Fluoromethylphosphoryl) oxypropane (commonly known as Sarin). The casualty list includes 12 people that were killed and thousands that were injured (Funato 2005). The Tokyo subway attack demonstrated the effectiveness of a small chemical payload in generating both physical casualties and mass hysteria. If the release of the chemical was coordinated with other simultaneous attacks, the responding agencies might be overwhelmed. A panicked population may affect both the efficiency of an emergency response and unnecessarily load the public healthcare system.

b. United States of America: September 11 Terrorist Attack (2001)

The September 11 attacks were a series of four coordinated terrorist attacks launched by the Islamist terrorist group al-Qaeda upon the United States in New York City and the Washington, D.C. areas on September 11, 2001 (NCTA, National Commission on Terrorist Attacks 2004) Four passenger airplanes hijacked by 19 al-Qaeda terrorists were intended to be flown in suicide attacks into targeted buildings. Of these planes, American Airlines Flight 11 and United Airlines Flight 175 were crashed into the North and South towers of the World Trade Center complex in New York City. Within two hours, the two towers collapsed. American Airlines Flight 77 (the third plane), was crashed into the Pentagon (the headquarters of the United States Department of Defense), causing a partial collapse to its western side. The fourth plane (United Airlines Flight 93) that was initially headed for the Washington DC area, crashed into a field near Shanksville, Pennsylvania. Almost 3,000 people died in the attacks, including all 227 civilians and 19 hijackers aboard the four planes. Of the three areas affected, the attack caused maximum havoc in the heavily populated urban environment. The search and rescue operation that ensued was one of the most complex and urgent operations ever conducted. Some unmanned systems and robots were used in this disaster (Murphy 2004).

c. Jakarta, Indonesia: JW Marriott Hotel Bombings (2003 and 2009)

The terrorist group Jemaah Islamiyah (JI) used a vehicular bomb in the execution of this incident. The incident involved an IED concealed in a mini-van on the periphery of the JW Marriott Hotel in South Jakarta. It resulted in the death of 12 people and injuries to 500 bystanders (International Crisis Group 2006). In 2009, the JW Marriott was again targeted by bombing attacks. Together with the Ritz-Carlton Hotel, both establishments were hit with bombings five minutes apart. The bombings resulted in 7 deaths (mostly foreigners) and more than 50 injuries to bystanders. These incidents present the vulnerabilities of private organizations collocated with the real targets of the perpetrators. This association is a cause for concern for private and public institutions alike and often means that collateral damage is unavoidable.

d. London, United Kingdom: Public Transport Bombing (2005)

The 2005 London bombings were an unannounced but a coordinated multi-effort attack, targeting civilians, distributed over both underground and surface levels (U.K. House of Commons 2006). The incident resulted in 52 dead and over 700 injured from four man-portable devices (homemade organic peroxide–based devices packed into rucksacks). Search and rescue operations were reported to be hampered by communication failings in which ground responders were unable to communicate via radio when deployed to underground areas (7 July Review Committee 2006).

Besides man-made disasters, natural disasters can also result in a need for urban search and rescue missions. These disasters usually span large geographical areas and in the case of an urban environment can cause a significant amount of damage. An unmanned vehicle, especially an UAV can provide invaluable information in establishing situation awareness and determining the areas or individuals which require immediate attention.

3. Urban Search and Rescue

Urban Search and Rescue (USAR) requires a multi-disciplinary organization that can perform physical, electronic, and canine search; extricate victims from collapsed structures; provide emergency medical care to victims and rescuers; assess and control affected utilities; perform hazardous materials monitoring; and evaluate and stabilize damaged structures. USAR task forces have to be equipped to respond to any type of disaster including man-caused intentional and accidental events (bombings, terrorism) and natural disasters (hurricanes, tornadoes, and earthquakes).

According to (Siciliano and Khatib 2008), unmanned rescue system could also potentially perform these series of task:

- **Search**: a concentrated activity in the interior of a structure, in caves or tunnels, or wilderness and aims to find a victim or potential hazards. The motivation for the search task is speed and completeness without increasing risk to victims or rescuers.
- **Reconnaissance and mapping**: broader than search. It provides responders with general situation awareness and creates a reference of the

destroyed environment. The goal is speedy coverage of a large area of interest at the appropriate resolution.

- **Rubble removal**: This activity can be expedited by robotic machinery or exoskeletons. The motivation is to move heavier rubble faster than could be done manually, but with a smaller footprint than that of a traditional construction crane.
- **Structural inspection**: to help rescuers understand the nature of the rubble in to order prevent secondary collapses that may further injure survivors to determine whether a structure is safe to enter. Robots provide a means of getting structural sensor payloads closer and in far more favorable viewing angles.
- **Acting as a mobile beacon or repeater**: to extend wireless communication ranges, enable localization of personnel based on radio signal transmissions by providing more receivers, and to serve as landmarks to allow rescuers to localize themselves.
- **Providing logistics support**: by automating the transportation of equipment and supplies from storage areas to teams or distribution points within the hot zone.

In the report compiled by Savannah River National Laboratory (Savannah River National Laboratory 2004), the Federal Emergency Management Agency and the National Institute of Justice identified the technology needs of USAR operations. The findings of the report highlighted ten areas of improvement, of which three areas could be greatly enhanced by implementing unmanned systems solutions. The three areas are:

- Improved real-time data access (data pertaining to site conditions, personnel accountability, medical information, etc.)
- The ability to communicate (transmit signals) through/around obstacles
- Reliable non-human, non-canine search and rescue systems - robust systems that combine enhanced canine/human search and rescue capabilities.

In a separate report (NIST 2005), the National Institute of Standards and Technology (NIST) discussed the statement of requirements for USAR robots performance standards. Using Defense Advanced Research Projects Agency (DARPA) categorization of robots, possible use of robotic/unmanned systems were highlighted and discussed. Of those categories highlighted, some applications that stood out were:

- Aerial high loiter systems that can Provide overhead perspective & situational awareness; provide Hazardous Material (HAZMAT) plume detection; provide communications repeater coverage
- Aerial rooftop payload drop systems that can deliver to rooftops; provide overhead perspective; provide communications repeater coverage
- Aerial ledge access systems that can conduct object retrieval operations from upper floors; crowd control with a loudspeaker object attached, provide situational awareness.

4. Urban Environment Unmanned Vehicles

In the Springer Handbook of Robotics (Siciliano and Khatib 2008), the authors highlights that unmanned systems in the USAR arena are an emerging technology, and have not yet been widely adopted by the international emergency response community. As of 2006, they have been used in some major disasters in the United States (World Trade Center, and hurricanes Katrina, Rita, and Wilma) (DeBusk 2009). However, recent advances have seen more adaptation of these technologies in operations. For example, several fire rescue departments in Japan and the United States routinely use small underwater robots for water-based search and recovery, a ground robot has been used for a mine explosion in the United States, and interest in the use of aerial vehicles for wilderness search and rescue is growing. The general lack of adoption is to be expected since the technology is new, and the concept of operations of novel technologies as well as the refinement of the hardware and software coevolution will take time. Rescue robot applications are relatively similar to military operations that the same platforms can be adapted for immediate use with little effort. However, some rescue tasks are significantly different than their military counterpart and are particularly unique to rescue. Additionally, the human–robot interaction for civilian response diverges from military patterns of use.

In the following section, some existing unmanned platforms that are suitable for urban search and rescue missions are presented.

a. DragonFlyer X4-ES

The DragonFlyer X4-ES is developed by Dragonfly Innovations Inc. and is the first commercially manufactured UAV to be federally approved for use by emergency services in North America (Saskatoon 2012). The model is sold directly to Public Safety Agencies. Its design features a four rotor configuration that is deemed ideal for small unmanned aircraft. The aerodynamics of the four rotor configurations inherently gives the system good stability, and allows it to achieve the best flight performance possible. Its construction is out of a rugged carbon fiber and injected nylon parts, ensuring durability and the ease of maintenance. Each system is equipped with a full suite of sensors that includes gyroscope, accelerometer, and barometric pressure sensor for state estimation.

Figure 4. Dragonfly Innovation Inc's Dragonflyer X4-ES (From Saskatoon 2012).

Besides the onboard sensors, the unmanned aircraft features a digital wireless video down-link, multiple camera payload options (including thermal forward looking infrared (FLIR) camera) with gyro stabilized camera mount, and handheld controllers available with integrated digital video display. The GCS controller provides real-time aircraft telemetry, camera control, on-screen live digital video feed, semi-autonomous flight modes for altitude hold and GPS position hold functions, map location display, and verbal warning messages.

During operations, the UAV is remotely piloted to the desired location where it can be locked in GPS position hold, automatically maintaining position and

freeing the operator to view the video displayed on the LCD screen. During flight the operator has the option to record video to the GCS, shoot still images, or network the GCS to other networks and stream real-time video to other devices. With such a capability, responders can obtain situational awareness without unnecessarily sending personnel into disastrous conditions. This dramatically reduces the risk to personnel and increases safety.

b. Variable Geometry Tracked Vehicle (VGTV)

Developed by Inuktun Technology, the VGTV is a miniature inspection system designed to access confined spaces and challenging terrain in a variety of applications, including search and rescue, nuclear and duct inspection (Inuktun 2012). The system's shape can be altered during operation to suit the physical conditions of the operating theatre. Being in their lowered configuration, the tracks take the shape of a tank allowing smooth transitions when travelling on flat surfaces. When the geometry is varied to the point where the vehicle is in its raised configuration, the tracks take the shape of a triangle. It remains fully operational throughout these shape alterations and can continue to travel and maneuver while its configuration is changing. This allows the vehicle to negotiate obstacles and operate in confined spaces and over rough terrain.

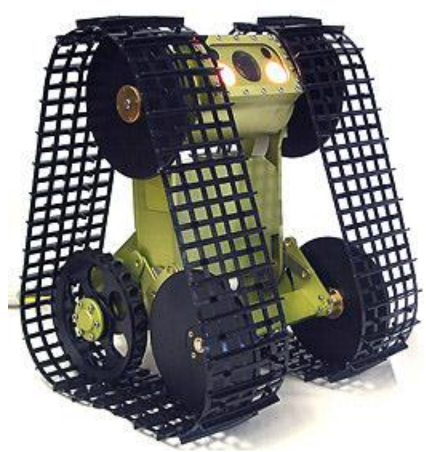


Figure 5. Inuktun's Variable Geometry Tracked Vehicle (From Inuktun 2012).

One of the limitations of the system is that it is a tethered system (tethered range between 100ft and 300ft). The vehicle is controlled using a hand-held control unit utilizing a joystick for speed and direction, and separate controls for the raise/lower, camera tilt, light and focus functions. It is relatively easy to operate and a new operator can learn to operate the vehicle effectively in minutes. The systems can also be built to incorporate a variety of sensors or devices required for specific applications (thermal cameras, ultrasonic sensors and etc.). In 2001, the system was used at the World Trade Center collapse in New York City for search and recovery work (Stopforth, Bright, and Harley 2010).

c. Maintenance Equipment Integrated System of Telecontrol Robot (MEISTeR)

The MEISTeR is a service robot created by Mitsubishi Heavy Industries. The system was first created due to the destruction of the Fukushima Daiichi Nuclear Power Plant (Mitsubushi Heavy Industries 2012). The Japanese robotic industry went on a quest of providing robots able to do dangerous work at the contaminated power plant. It is a two-armed robot designed to assist recovery work after disasters or severe accidents by performing light-duty tasks in areas inaccessible by humans.

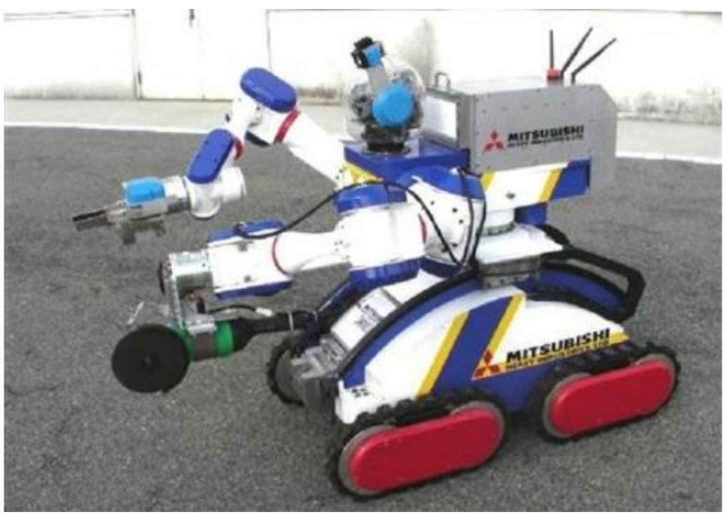


Figure 6. Mitsubishi Heavy Industry's MEISTeR (From Mitsubushi Heavy Industries 2012).

Its robotic arms have seven degrees of freedom and can lift a load of 33 pounds each and it is robust enough to withstand the highly radioactive environment. By changing its arms' attachment tools, the robot can perform various tasks such as carrying objects, drilling and opening/closing of valves, clearing obstacles, and piercing through concrete to check radiation levels. It can move at up to 1.24 mph and negotiates uneven terrain, including stair steps up to 8.5 inches high on its four independently moving tank tracks. This current model is optimized to perform under the conditions of the Fukushima disaster but can be altered to suit the needs of other types of disasters. In essence, it is a system that is capable of being a surrogate for human operators. Additionally, it can also reduce the responders' burden by carrying additional equipment and tools necessary for the rescue effort.

d. VideoRay Remotely Operated Underwater Vehicle (ROV)

The VideoRay ROV is an underwater submersible remotely operated underwater vehicle (VideoRay 2012). When using the system remotely, the operator is able to get live video feed from the onboard camera and provide a sense of the situation underwater. Using attachments such as scanning and imaging sonars, positioning systems and manipulators, the ROVs carry out port security and ship inspections and deep-sea wreck location, and search and rescue operations.

Figure 7. VideoRay's ROV (From VideoRay 2012).

While most disasters are associated with large scale search and rescue activities on land, inspection of coastal littoral regions is also important. Bridges must be inspected, as they are needed for responders and recovery workers to have access to affected areas, and because they influence the general recovery of the area. As highlighted in (Steimle et al. 2009, 1–6), man-made features (e.g., seawalls, levees, or dikes) may be compromised during a disaster and can possibly lead to a secondary disaster such as seen in New Orleans at Hurricane Katrina. Thus, the channels must be restored and docks repaired as part of the economic recovery and restoration of shipping. In the inspection operation (Steimle et al. 2009, 1–6) performed by the team, it was found that the role of humans were vital in achieving operational objectives. Human activities include piloting, operating the sensors, interpreting the data, and providing safety oversight. A minimum of four to six people were involved in the operations at any time.

Currently, the United States Army Corps of Engineers, U.S. Immigration and Customs Enforcement, New York Police Department (NYPD), Port of Long Beach Harbor Patrol, and fourteen branches of the Maritime Safety and Security Team (MSST) of the United States Coast Guard are among its users.

5. Challenges of using Unmanned Systems

Various literature sources (Murphy, Stover, and Choset 2005; Nguyen et al. 2009; Siciliano and Khatib 2008; Stopforth, Bright, and Harley 2010; Trivedi 2011) highlight the current challenges faced by the designers and operators alike. In this section, some of the most obvious challenges are listed and discussed. Additional challenges, including integration are discussed later in Chapter II of this thesis.

a. Mobility

For all modes of unmanned systems, mobility remains a major problem. This is especially relevant for ground vehicles used in urban search and rescue operations. The complexity of the environment caused by the unpredictable combination of vertical and horizontal elements, coupled with the dynamic environment provides a stern challenge for vehicular mobility. Actuation and mechanical design needs to be improved to provide the unmanned systems with the capability to move through the

obstacle with ease. While ground vehicles are susceptible to terrain obstacles, aerial vehicles are concerned about their cluttered urban environment too. Power lines, buildings and vegetation provide challenges to their mobility.

b. Communication

Most unmanned systems require constant communication with its operators for successful operations. It enables the operator to sense and see what the system picks up. Constant communication is carried out either with a tether or via wireless radio. Due to imagery requirements for a search and rescue operations, systems usually have high bandwidth demands. Additionally, for optimal operations and control, latency of the communication has to be low. For systems to be truly unmanned, the wireless mode of communication is preferred. However, in a disaster aftermath where communication infrastructure is disrupted and where systems are required to be deployed into underground or indoor environments, wireless communication is problematic. The dynamic environment interferes with the propagation of radio waves and affects optimal communication.

c. Sensors

Without the ability to sense adequately and persistently, the rescue systems would lose its capability as a mission asset. For instance, the rescue system might be close to a casualty and not be able to sense the presence of the casualty. Besides the capability of the sensor, the physical size of sensors also impacts its usability. Payloads that are too large are not suitable for unmanned systems operating in confined environments. Large payloads will also tend to deplete the power of the system more rapidly. There is a trade-off between the size and capability of the sensor. Sensing algorithms are also required for the interpretation of sensing data. The interpretation of these data needs to be intuitive to human operators in order to permit effective operations.

d. Power

Power subsystems for unmanned systems generally takes two forms; batteries or internal combustion engines. The former is preferred over the latter due to

logistical and safety implications. The size and weight of the power source is important and it is certainly one of the drivers of platform size, and can affect the placement and overall capability of its payloads. The design of the battery placement also needs to be considered, it should be protected from external elements, and yet it has to be easily replaceable with minimal efforts.

e. Integration

The systems involved in a USAR operations need to be integrated to perform its required mission. These systems include the central control station, the ground control station, the unmanned systems, the ground operators and the relay UAVs. They come together to operate as a "system of systems" and facilitate capabilities that a single system could not achieve. According to the International Federation of Robotics (IFR), the rate of growth in the use of corporative unmanned system is expected to increase to 17 percent by 2013. The experience of the past several years also shows that the impact of special robots is very minor because individual devices and systems often cannot function with one another in the field (Robotic Trends News Source 2011).

6. Optimization using the Inverse Dynamics in the Virtual Domain methodology

Rescue missions are typically fifteen minutes, and it is usually flown within the vis_ual field. Additionally, flight in the congested urban environment requires agility and accuracy. Thus, traditionally waypoint navigation does not quite make sense for operations in an urban search and rescue operations. The current available option for such operations would be to utilize manual control. Manual control would require at least three personnel to operate the vehicle (Murphy and Stover 2006) and thus may not be able to realize its full potential in a resource-limited rescue operation. The ideal would be to shift workload away from the human operators by making the system autonomous. In order for an unmanned system to be fully autonomous, an appropriate controller that incorporates both the roles of trajectory planning and trajectory following is required. The two domain of autonomy are usually tackled separately and there exist a substantial amount of

literature on both domains (Hargraves and Paris 1987, 338–342; Jwa and Ozguner 2007, 567–580; Murphy, Stover, and Choset 2005; Richards and How 2002; Valenti et al. 2006; Cowling, Whidborne, and Cooke 2006).

The optimal control problem this thesis addresses is to guide an aerial system from an initial state to some final state with constraints imposed on both the states and the controls. Ideally, the routine that is used should be capable of updating itself multiple times over the course of the trajectory to mitigate disturbances and un-modeled motor dynamics. A great amount of research has been done with regard to this type of optimization and there are a wide variety of optimization software packages (such as OTIS (Paris and Hargraves 1996), SOCS (Betts and Huffman 1997), DIRCOL (Ross 2004) and DIDO (von Stryk 2000)) available that can be used to solve problems somewhat quickly. However, many of these methods, such as the direct collocation method, rely on thousands of variables and constraints, which add significantly to computational time and cost. Traditional indirect methods are not able to handle this problem in real-time, leaving the alternative direct method as the ideal choice to formulate the vehicle's path. In "Direct Method Based Control System for an AutonomousQuadrotor" (Yakimenko et al. November 2010, 285–316), the authors explained that for some unmanned systems with relatively long mission time, using established optimization software, the platform achieves a real-time computation speed of a few minutes. However, for systems that have mission time of a few minutes, the aforementioned real-time computation methodology would not suffice. The proposal is for a direct method based optimization methodology that can achieve the required computation speed in latter scenario.

In their journal article (Yakimenko et al. November 2010, 285–316), Yakimenko et al. propose a real-time control algorithm for autonomous operation of a quadrotor unmanned air vehicle. They found the quadrotor platform to be a small agile vehicle, making it suitable as an excellent test bed for advanced control techniques. It was proven to have useful applications in various areas including internal surveillance, search and rescue and remote inspection. The proposed control scheme incorporates two key aspects of autonomy (i.e., trajectory planning and trajectory following.) The key feature of the

control method introduced by the paper was that it uses inverse dynamics where the state and control inputs can be expressed as functions of the output and its derivatives can be termed “differentially-flat.” As a result, all optimization occurs in the output space as opposed to the control space. The speed profile was optimized independently. This is followed by the mapping it to the time domain via an identified speed factor. With the identification of the optimized trajectory, the trajectory following portion is finally perfomed by a standard multi-variable control technique. In the event of mission changes or conflict, a digital switch is also incorporated to re-optimize the reference trajectory.

Martin, in his thesis (Martin 2012) presents the coordination of an unmanned, multi-vehicle team that navigates through a congested environment. In the absence of global positioning data, a series of sensors were utilized to achieve localization. The quadrotor used was equipped with a downward-looking camera and sonar altimeter, while the ground vehicle was equipped with the sonar and infrared range finders. The mentioned equipment, when used in tandem with an infrared-based motion-capture system, effectively provided the required indoor localization capability. By using conventional image-processing techniques, the quadrotor used in the experiment were able to generate bird’s-eye images providing information about dynamic environment to the ground vehicle. The ground vehicle then generates a suitable trajectory and avoids the identified obstacles. From the experiments performed, he concluded that the direct method based trajectory generator was able to generate a collision-free path that optimally brings the ground vehicle to its final destination. He also highlighted that the this method of trajectory optimization/generation is faster than other indirect methods.

In his thesis (Chua 2012), Chua explored the various types of UAVs deployed for urban operations and investigated the trends of the UAVs in terms of their parameters such as weight, altitude, speed, and sensor suite. The challenges and requirements for interoperability of multi-UAV operations in urban environments were also discussed. A direct-method-based control system for multiple UAV collaboration and obstacle collision avoidance was proposed (Chua 2012). The UAVs were able to share and integrate their sensors’ information for joint cooperation. A dynamic model was developed for the simulation testing of the algorithm. Testing his model, he demonstrated

that two UAV could perform non-centralized guidance and control by generating quasi-optimal trajectories and avoiding obstacles. He noted latency in the experimental model but highlights that the model has sufficient safety margin for a collision-free flight.

In both the theses mentioned earlier, the trajectories generated using the IDVD methodology were restricted to motions in the horizontal plane. This thesis will explore the ability of the methodology to generate trajectories that span both the horizontal and vertical planes of travel.

THIS PAGE INTENTIONALLY LEFT BLANK

II. SYSTEMS ENGINEERING CONSIDERATIONS

A. PROBLEM DEFINITION

It is evident that with increasing urban population, the urban environment is fast becoming an important operating arena that authorities should focus on. When a disaster (natural or manmade) strikes an urban environment, a swift search and rescue operation improves the probability that survivors can be rapidly located and rescued. In order to do that, the responding agencies would first need to have substantial situational awareness of the area of operations. Situational awareness is the comprehension of the environmental elements with respect to time and space, and their projected change over time. Continual surveillance of the disaster areas is also critical in ensuring mission success. The chaotic aftermath of a disaster coupled with the complex urban environment makes acquiring situational awareness a daunting task.

Besides situational awareness, there is also a requirement to satisfy the logistical demand of the operations. Having enough supplies and resupplies for the responding personnel is vital to the successful completion of the rescue operations. Due to the unpredictable and unstable nature of the aftermath, sending another responder to the scene for resupply might not be feasible. The terrain could be hard to navigate and precious time would be lost. In addition, it could be unsafe for both the initial responder and the responder bringing the supplies. A faster and safer method is required.

Effective inter-agency cooperation is also required for successful operation. With a seamless flow of information, resources can be channeled to the area of operation that is in dire need of support. Operating in an urban search and rescue environment, the stakeholders of the system requires a system that can reliably provide superior situational awareness (mapping, changes, target detection and responding personnel location.) Logistically, the availability of supplies and resupply is vital to the success of the mission. Therefore, the ability to deliver supplies to the frontline is an important feature of the system.

B. BOUNDARIES

There are multiple considerations when operating in an urban search and rescue environment. One can appreciate the intricacies of operating in such environment by studying the boundaries of this environment. Boundaries can be either physical or imaginary. In essence, the boundaries bound each object, either tangible or intangible, into discernable entities. There are three basic boundary classifications; Physical, Functional and Behavioral. These boundaries can impart limitations, constraints and other boundary conditions to a system. Identifying problem boundaries can reveal added dimensions within the problem space.

1. Physical, Functional and Behavioral Boundaries

The physical, functional and behavioral boundaries of operating in an urban search and rescue environment are presented in Table 1. Physical boundary includes the physical objects in the environment. Functional boundaries are formed at the interfaces of objects and exposed the interaction between objects (Langford 2012). Finally behavioral domain exists due to the existence and interaction of physical and functional domain.

Table 1. Physical, functional and behavioral boundaries

Physical Domain	Functional Domain	Behavioral Domain
a) Buildings b) Debris c) Vehicles d) Roads/highways e) Emergency Responders f) Civilian Volunteers g) Casualties/Survivors h) Rescue Equipment	a) Emergency Response Process b) Evacuation Process c) Search Process d) Multi-Agency operational process e) Triage Process	a) Civilian reaction b) Responder preparedness c) Multi-agency Interaction d) Emergency Preparedness

2. Input-Output Model

By utilizing the listed boundaries in the previous section, the input-output diagram was derived. The input-output analysis provides a primary sense of the system interactions with its environment. Inputs to the system can be segregated into controlled and uncontrolled input. As the name suggest, controlled inputs are the energy, matter,

material wealth and information (EMMI) directed into the system to aid system operations. Controlled inputs include the responders, the rescue equipment utilized, and situational information available for planning. On the other hand, uncontrolled inputs are usually EMMI that the responders have direct control, and are occurring due to the nature of the operations. For instance, the cluttered operating condition in a USAR operation is not a controlled input, but a result of the circumstances. Similarly, when the incident occurs in an urban environment, the mass casualty situation is not a controlled input.

On the other side of the Input-Output diagram are the outputs of the system. It is again segmented into two types of output, namely intended output and unintended output. Again, the description of each term is self-explanatory. Figure 8 presents the Input-Output diagram for the search and rescue operations in an urban environment.

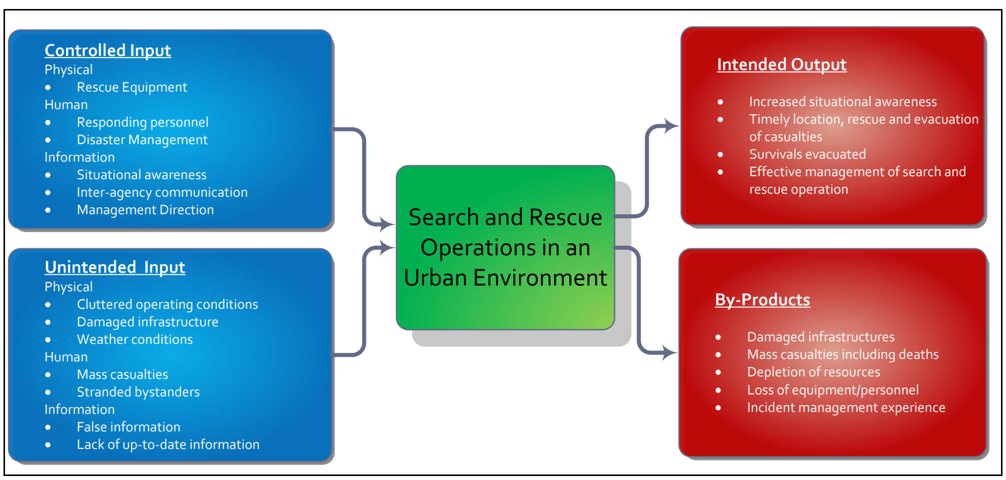


Figure 8. Input-Output diagram

3. Contextual Diagram

The system context diagram is a diagram that represents entities outside of the System, which could interact with the system (Kossiakoff and Sweet 2002). Through the development of the context diagram, stakeholders that were initially less visible became apparent. The diagram also identifies the interaction of these entities with the search and rescue operations. The contextual diagram for the system is depicted in Figure 9.

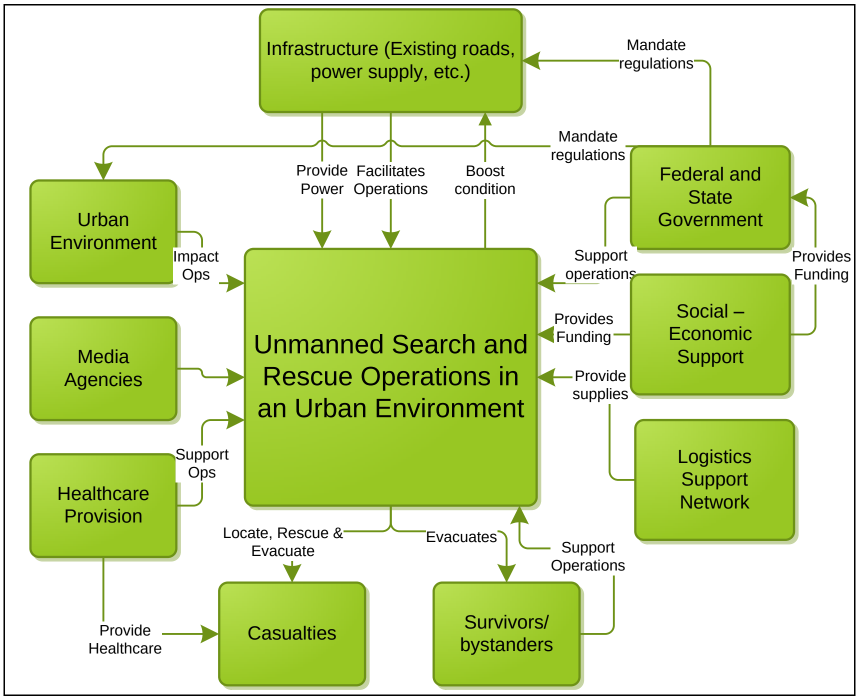


Figure 9. Contextual diagram for the system

The main entity of the contextual diagram in focus is the unmanned search and rescue operation in the center of the diagram. The other entities includes elements such as the urban environment that the operations will be performed, the existing infrastructures, the Government, social-economic support groups, the general public, healthcare provision support, and the media agencies. The urban environment, as presented in Chapter I, presents a real challenge to unmanned search and rescue operations. The urban environment impact the operations by presenting a set of challenges (e.g., line-of-sight linkages, airspace limitation, dense built-up environment and etc.) that the unmanned systems need to overcome. It is important for the operation to be cognizant of the urban environment's impact.

Both the urban environment and the existing infrastructure are dependent on the government for rules and regulations. These rules and regulations determine how the urban environment is structured and also how the infrastructures are developed. Existing infrastructure provides the necessary support for operations in terms of facilities, utilities

and the general condition of the operations scene. The government is also important in delegating resources to the operation scene to alleviate the situation for the responding forces. The support could come in terms of manpower and funding of the operation.

The next entity is the general public. One of the aims of the operation would be to evacuate the affected populace to a safer environment. However, the high population density in urban areas could prove to be challenging to evacuate and it is important to systematically carry out this process to avoid unnecessary casualties due to evacuation. On the flip side, the general public is also a good source of support for the operations. Support could come in the form of the public volunteering their services, or in the form of monetary funding. Social economic groups representing the general public can also contribute positively to the operations by providing monetary and equipment support.

The healthcare provision system would have to be ready to respond to a mass casualty situation. They must be able to receive and treat a large number of casualties that is expected from an urban disaster. The ability of the healthcare system to receive and treat casualties impacts the success of the operation. Depending on the magnitude of the disaster, the healthcare system may also delegate valuable resources to the disaster site to aid ground operations.

Media agencies interact with the system by streaming the actions and information from the ground and presenting it to the general public. This aids the spread of vital information to the populaces who are not directly involved in the disaster and can provide some beneficial input to the operations. First, the public would know to avoid the disaster area and not cause added strain to the operations. Second, depending on the magnitude of the disaster, information that is broadcast internationally may invite help from other international organization that specializes in urban search and rescue.

On top of identifying other stakeholders, the analysis also surfaced limitations and constrains of the system. Limitation, constraints and the stakeholders are discussed in the next section.

C. LIMITATION AND CONSTRAINTS

Limitations and constraints are the boundary conditions that results from the identification of the boundaries in the previous section. Identification of the limitation and constraints further contributes to the crystallization of the problem space. The following sections list the limitations and constraints from the perspective of a systems engineer. The respective impacts on the system were also discussed.

1. Limitations

In an urban environment, a robotic system will need to traverse seamlessly between both environments while maintaining accurate localization and effective mapping. One of the most glaring limitations of operating traditional unmanned systems in an urban environment is the limited line of sight with the satellite and the ground control station (GCS). The termed "urban canyons" are used to describe urban environment where Global Positioning Satellites (GPS) signals are lost or unreliable. Besides GPS navigation, other localization techniques could be coupled together to maximize the accuracy of localization. Additionally, multipath propagation effects could cause the signals of the unmanned systems to attenuate and become ineffective. There is a need to identify the exact limitation and provide a solution to overcome it.

Due to the cluttered nature of the operating environment after a traumatic event, the amount of obstacles that the unmanned system is required to overcome is tremendous. Many of these obstacles are dynamic in nature and can provide some of the most challenging environment for the system to navigate. As much as possible, prior information should be used to enhance the system capabilities. Failing which, the system must be adept in obstacle avoidance and maneuvering in the cluttered environment.

Technological and resource limitations can limit the capability of the unmanned systems utilized resulting in a less capable platform. Capability gaps may not be sufficiently met by the resource and technological limitation of the organization. However, the best system available would be one that the responding organization can effectively procure, manage and operate. Besides the operating hardware, the life-cycle aspect (spares and maintenance support) has to be considered in parallel.

Human-robot interaction can also impart limitation to the unmanned system. If an operator cannot use the system fluently, then the effectiveness of the platform is questionable. Currently, operators would have to be extensively trained to be proficient with using the system. Thus the ability to train and provide proficient operators is one limitation of the system that can be improved. For operations, the human-robot ratio is also an issue. With current technology, such a system still requires two to three operators. The effectiveness of the system depends on the improvement of this ratio. User interfaces are a key component in providing interaction and situational awareness for the human operator. Thus, the usability and functionality of the user interface is a limitation to the system.

2. Constraints

Constraints for the system could arise from regulatory authorities (such as the airspace management) constraints the unmanned system to operate in a specific and confined space. Although cumbersome, it is an important constraint as the urban operations area is a crowded environment. Having airspace de-confliction assures a safer operating environment for both unmanned and manned vehicle.

The system would also have to work under the constraints of safety protocols and regulations. In the highly dynamic urban environment, there exist many hazardous conditions for both the responders and the member of public. Safety constraints placed upon operations will safeguard the interest of the personnel involve. For instance, certain no fly over zones may be imposed on areas where masses of personnel or responder congregate. Although the constraint may impact the effectiveness of the system, the trade-off between safety and effectiveness is justifiable.

The hazardous nature of the operating environment can constraint the characteristics of the unmanned system. Its communication capability, mobility, sensor suites, power capacity are all dependent on the environment it will be operating in.

D. STAKEHOLDER ANALYSIS

A stakeholder is any owner and/or bill payer, developer, producer or manufacturer, tester, trainer, operator, user, victim, maintainer, sustainer, product improver, and de-commissioner. Each stakeholder has a significantly different perspective of the system and the system's requirement (Buede 2009). This section describes the stakeholders identified for this system and describes their needs.

1. Casualties

As the main purpose of the search and rescue mission is to recover as many victims from the disaster as possible. Mass casualty situations occur when the number of casualties exceeds the available medical capability to rapidly treat and evacuate them. In disaster relief operations and in the aftermath of terrorist incidents, mass casualty situations frequently occur (Federal Emergency Management Agency 2012). The main need is for casualties to be removed from the hazardous environment. Besides their main need, casualties also need to be given immediate care and treated for their conditions. Casualties are usually classified into four distinct categories:

a. Minimal Attention (Green)

These casualties have suffered no impaired functions and can either self-treat or be cared for by non-professionals. Some of their conditions include abrasions, contusions and minor lacerations. Their main need would be to receive medical supplies and attention that will help alleviate their injuries within a 24-hour time frame.

b. Delayed (Yellow)

The "delayed" category of casualties is a group where advance medical care can be delayed upon administration of simple first aid. These casualties require further medical attention but can reasonably be expected to remain stable if medical attention is delayed. They need to receive treatment within a 4-hour timeframe.

c. Immediate (Red)

Casualties that require immediate attention are those that are seriously injured and can be reasonably be expected to survive if immediate medical attention is rendered. They usually suffer from an obvious threat to their lives or limbs, or complications in their life sustaining functions. Medical attention should be given to these casualties immediately within a 2-hour timeframe.

d. Expectant (Black)

The last group of casualties consists of expectant or deceased victims that have injuries that are so extensive that even if they were the sole casualty and had the benefit of optimal medical resources application, their survival would be very unlikely. They include victims who are unresponsive with no pulse, or individuals with catastrophic head and chest injuries.

2. Responding Agency

Responding agencies are the organizations that are tasked to respond to disaster sites to perform their duties in an urban search and rescue operation. They include law enforcement agencies, fire-rescue, emergency medical service, and military responders. In many disasters, volunteer search and rescue organization would render their services in support of the response. Besides performing rescue operations, responders would need to manage the rescue operations that include situational awareness, manpower management, emergency medical management and logistical support for the operation. Disaster Response personnel are trained, competent and able to perform their duty. They need the support provided by their respective organization and each other to perform their roles efficiently. The safety of the personnel involved in the operations is also an important consideration.

3. Government

The government of the disaster-stuck nation is a key stakeholder. It will set the policies, rules, and regulations of the nation. These include the unmanned systems operating template (in the case of a UAV airspace restriction), the standards for the urban

environment (e.g., building codes that influence building materials used, safety and building height; road width and etc.) The general state of the responding agencies is also a direct responsibility of most government. The responding agencies are usually state funded entities that provide public services. Additionally, the effectiveness of the government in managing disaster would be an important performance indicator to the general public and observing parties. It will want to portray efficient emergency management ability.

4. General Public

The general public is the group of stakeholder indirectly affected by the disaster. Without being physically involved in the incident, they can be emotionally affected by the incident. They will be monitoring the events of the incident and can provide assistance monetarily or by volunteering their services. They will also judge the operational effectiveness of their government and responding agencies in the rescue efforts. The response of the general public to the incident could be at an individual, community or societal level, with the latter having the most influential response.

5. Business Owners/ Commercial Entities

Commercial entities can also be generous supporters to disaster relief and can contribute substantially to the success during and after the incident. In the recent Hurricane Sandy disaster, more than U.S.$400 million goes to relief and disaster recovery efforts (Center for Disaster Philanthropy 2013). These sources of assistance can be invaluable to the disaster-hit area. On the other hand, business owners are also affected when a disastrous incident occurs. The properties they own and their commercial viability are adversely impacted by the incident. One of their main concerns would be to recover from the situation as quickly as possible, minimizing losses due to down time.

E. EFFECTIVE NEEDS AND CAPABILITY REQUIREMENTS

Various stakeholders defined in the earlier sections have different needs that are inherent to the system. Studying their needs results in the formulation of a series of effective needs required for the system.

Being able to make important decisions based on timely and accurate ground information is of high importance in managing disaster. There several categories of urban search and rescue operation missions: Fire-fighting and rescue, emergency medical response, surveillance, logistical support and perimeter/site management. Each category has common and unique needs. Besides the core missions, personnel management is also an important function of the system. It is desirable to have systems that can provide higher autonomy. This can reduce the number of operators for routine task and optimize the limited manpower. The various operational capabilities requirement based on the perspectives of the stakeholders for an USAR operation are summarized as:

- Provide situational awareness and disseminate/receive valuable information to responding agencies
- Effective management of limited resources
- Timely response to casualties that require assistance
- Provide timely intelligence and information to support decision making
- User interface for unmanned system should be intuitive and functional
- Provide logistic support to the personnel on ground

F. PROBLEM SCOPE

The scope of work for the System is to work within the bounds of the defined boundary, the limitations and constraints and the stakeholder analysis, defined in the previous sections. These considerations make up the scope of this study, while others are outside. This effort provides focus and ensures a solution set that will enhance the capabilities of the system.

1. Within Scope

The cluttered urban environment provides a challenging environment for the unmanned system to operate in. The ability to maneuver beyond the obstacles while carrying out its mission is critical and is therefore within the scope of this thesis research. Arriving at a feasible obstacle avoidance strategy is paramount to the ability of the unmanned system to carry out its mission. The chosen platforms to be used in the operating environment should also be able to perform capably (For e.g., fixed-wing UAV

might not be suitable for operations due to its inability to hover or fly at low speeds). The speed of responding to the casualties is of utmost importance and should be carried out speedily without destabilizing the delicate nature of a post-incident site. By utilizing unmanned systems, more personnel will be available to perform vital search and rescue, and medical provision operations.

2. Outside Scope

This thesis will not include the use of unmanned systems in the prevention and preparedness phases of disaster management. Although UGVs, USVs and UUVs are effective and are used in rescue operations, they will not be the focus of this thesis. The focus of this thesis is limited to the generation of a feasible obstacle avoidance method for UAV platforms. Additionally, post disaster management efforts will not be in the scope of this thesis.

G. OPERATIONAL CONCEPT

Ultimately, the system should be able to provide a solution to meet the specified capability requirements. The concept of operations for providing a search and rescue capability in an urban search and rescue environment can be illustrated by Figure 10. A central command station (CCS) acts as the nerve center for the operations. It will be tasked with the role of resource coordination and decision-making command. The CCS communicates to other ground control centers and responding personnel via a series of relay UAVs. This provides the primary form of linkages. These relay UAVs also provides the other unmanned systems with vital positional information.

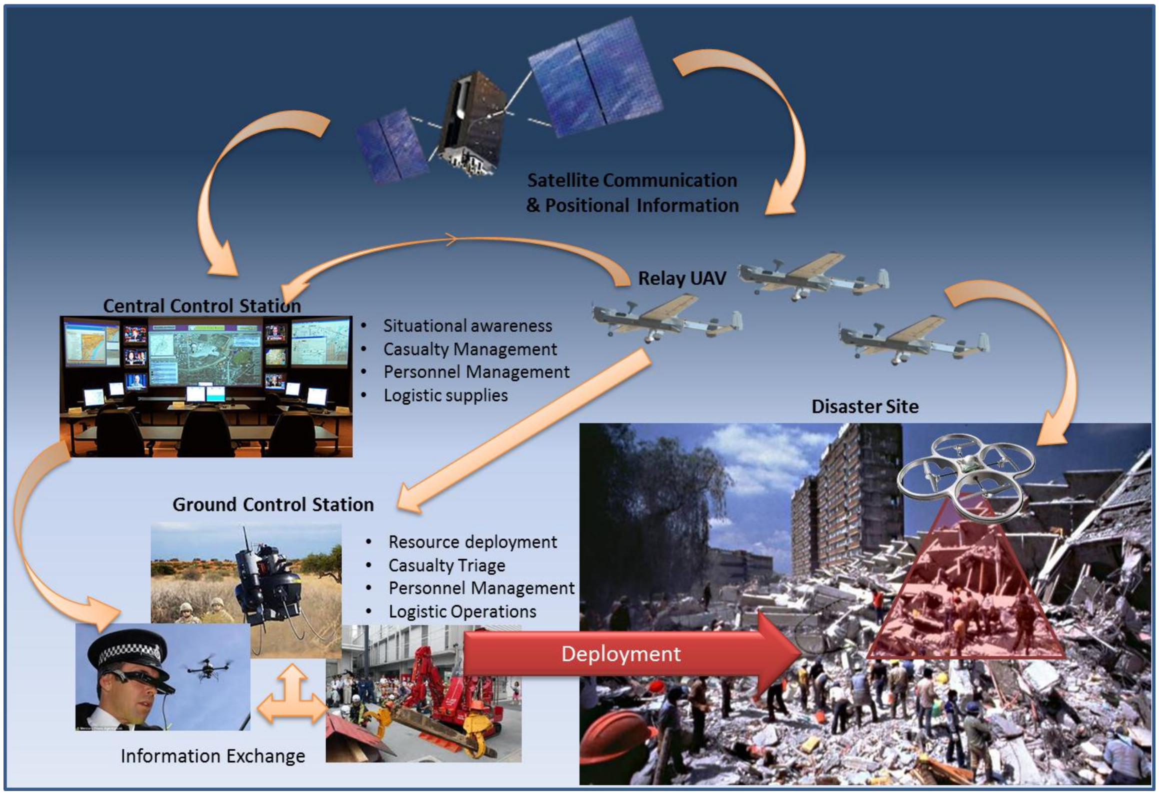


Figure 10. Concept of operations

With this setup, the CCS does not need to be in line of sight of the ground assets or personnel. Personnel and ground assets will provide vital mission data (e.g., mission area, most recent map data, number of unmanned assets and situational awareness reports, updated casualty location, personnel location, etc.) Ground assets, controlled by a network of ground control stations (GCS), include a series of unmanned vehicles (UAVs, UGVs) and the responding personnel. The unmanned systems are equipped with sensors that can gather information of the disaster site. Besides persistent surveillance, specialized unmanned systems can also perform roles of casualty search, rubble removal, structural inspection, mobile repeater or a surrogate for responding personnel. From the information obtained, the CCS has an acute sense of situational awareness and can direct ground assets efficiently.

With the information received, GCSs can perform resource allocation and plan the general waypoints for their respective UAV/UGV to follow. It then sends the waypoints as general guidance to the unmanned vehicle to carry out operations. The systems are

equipped with a real-time trajectory generation based on IDVD (Yakimenko et al. November 2010, 285–316), imparting it with the capability of performing dynamic retargeting and obstacle avoidance. This is vital as operations in a dynamic site and rapid changes to the environment are expected. Multiple unmanned systems have the capability to work together seamlessly. For instance, when a UAV equipped with deep scanning equipment discovers the position of a survivor, the UGV station would receive the location information and proceed to extricate the rubble, allowing rescuers to reach the casualty. Other than deep scanning capabilities, the unmanned systems are also delegated other auxiliary functions that will have them performing structural inspection, doubling as communication relays, or by providing autonomous logistics support to forward-deployed rescuers.

H. FUNCTIONAL ANALYSIS

The functional architecture of a system contains a hierarchical model of the functions performed by the system, the system's components, and the system's configuration items; the flow of informational and physical items from outside the system through transformational processes of the system's functions and on to the waiting external systems being serviced by the system; a data model of the system's items; and a tracing of input/output requirements to both the system's functions and items (Buede 2009). In this section the functional analysis of the system will be performed to derive the functional decomposition of the system.

By performing a functional analysis, the vital functions of a search and rescue mission in an urban environment can be defined. Gathering the information from the previous sections, the pertinent "high-level" functions were defined and further developed with "derived" functions. The functional decomposition of the functions and its description can be found in Table 2. Additionally, the functional hierarchy is also presented in Figure 7 to provide a graphical relationship between the functions.

Table 2. Functional decomposition and description

S/N	Function	Description
0	Unmanned Search and Rescue Operation in an Urban Environment	The top level function of the system
1.0	Provide Situational Awareness	The system level function of providing situational awareness across the operation
1.1	To monitor situational developments	Monitor the situational development of the ground via inputs from the various assets on ground
1.2	Update central control station	Update the Central Control Station and provide the most up to date information for resource allocation and decision-making
1.3	Update ground assets	Update the Ground Assets and provide the most up to date information for ground operations
2.0	To communicate	The system level function of providing communication between various system assets
2.1	Provide communication between CCS and ground assets	Provision of a means of communication between the CCS and ground assets
2.2	Provide intercommunication between ground assets	Provision of a means of inter-communication between the ground assets
2.3	Provide communication between relay UAV and all other stations	Provision of a means of communication between the relay UAV and all other stations
3.0	To maneuver	System level function of providing the ability for the assets to maneuver on the ground
3.1	Sense environment for obstacle	Provide the ability to sense for obstacle in the dynamic environment
3.2	Avoid obstacles (Terrain & friendly forces)	Provide the ability for the assets to avoid contact with terrain, buildings or friendly forces
4.0	To extricate casualties	The system level function to extricate and retrieve casualties from the operations
4.1	Detect casualties	Provide the ability to detect the presence of casualties
4.2	Remove obstacles	Provide the ability to remove obstacles that prevent access to the casualty
4.3	Retrieve Casualties	Provide the ability to remove the casualties from harm
5.0	To provide logistical support	System level function of providing logistical support to ground operators
5.1	Provide communication support	Provide a means to enhance the communication capability of the ground troops (e.g., communication Beacon, communication surrogate etc)
5.2	Provide equipment support	Replenishment of supplies and support equipment to the ground operators

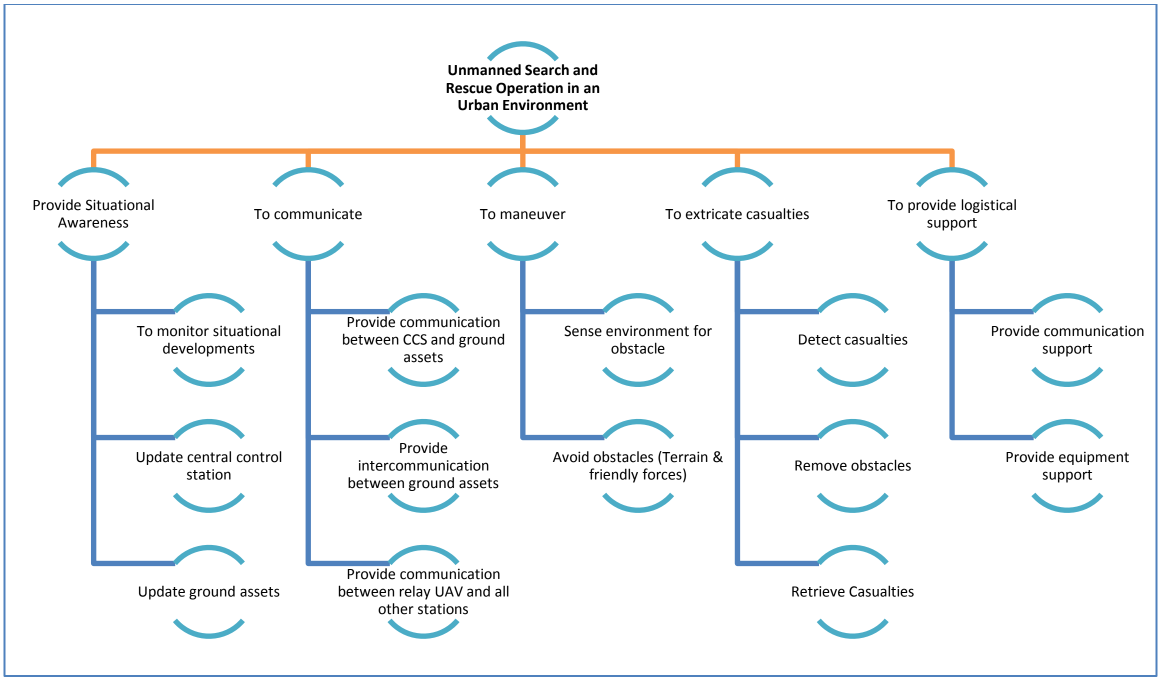


Figure 11. Functional hierarchy of system

I. INTEGRATION CHALLENGES

Systems integration is the unification of the objects and their interactions of energy, matter, material wealth, and information (EMMI) to provide system level functionalities and performances. Integration is a method that facilitates outcomes that are beyond what an individual object can do either individually or by a number of objects acting independently (Langford 2012). Integration involves the actual adoption of ideas and changes as opposed to Interaction where potential for integration exist. The systems involved in the USAR operations need to be integrated to perform its required mission. These systems include the central control station, the ground control station, the unmanned systems, the ground operators and the relay UAVs. They come together to operate as a "system of systems" and facilitate capabilities that a single system could not achieve. However, there are several integration challenges that need to be overcome, and they are listed in the following section.

1. Communication

Most unmanned systems require constant communication with its operator for successful operations. It enables the operator to sense and see what the unmanned system does. These constant communications are carried out either with a tethered or via wireless radio. Due to imagery requirements of a search and rescue operations, systems usually have high bandwidth demands. Additionally, for optimal operations and control, latency of the communication has to be low. For systems to be truly unmanned, the wireless mode of communication is preferred. However, in a disaster aftermath where communication infrastructure is disrupted and where systems are required to be deployed into underground or indoor environments, communication is problematic. The dynamic environment interferes with the propagation of radio waves and affects optimal communication.

2. Sense Making

One of the key challenges is how to localize, map, and integrate data from the unmanned systems into the larger geographic information systems used by strategic decision makers. Sensor data fusion can be achieved by either gathering sensor data from different sensors or receiving sensor data from multiple unmanned systems and combining them into improved data. At Ohio State University, research was conducted at that utilizes layered data fusion from multi-UAV sensing. An information theoretic cost function was applied and a cooperative optimization method on multiple mini-UAV sensing was performed. This cooperative mission was achieved via the alignment of two layered video sequences (Jwa and Ozguner 2007, 567–580). Professors Oleg Yakimenko and Gerard Leng suggested another method to perform continuous surveillance of a target in an urban environment, using multiple fixed-wing UAVs with a formation flight control algorithm (Leng and Yakimenko 2011). To track an object, the unmanned systems require an unobstructed line of sight. Given the congested operating environment, maintaining a constant line of sight might not be achievable. Thus, deployment of multiple unmanned systems can share vital information that is required to have a constant track on the target.

3. Human-System Interactions

Unmanned systems are part of a human-centric system; even if systems were fully autonomous, responders require the information in real time and need the ability to dictate and modify the system's tasks. Four key problems of Human-Machine interactions need to be resolved. Firstly, the human-to-system ratio for operations needs to be improved. A single unmanned system requires between two and three humans, depending on modality, for safe and reliable operation (Murphy and Stover 2006). Second, these operators have to be extensively trained, which may be out of the question for many response teams. Rescuers have limited time compared with military operators to train extensively with unmanned systems and have lesser opportunities to perform operations. Thus training operators to become competent users is a challenge. Third, providing situational awareness is a vital function and current systems do not suffice and need further improvements to achieve seamless integration. This will in turn reduce training demands. Finally, there is a need for affective robotics due to the fact that other personnel will interact with the robot outside of the operator role (Siciliano and Khatib 2008). The "10 min rule" of human-computer interactions states that if an operator cannot use the main functions of the system within 10 minutes (Nelson, T. 1980), there exist some inherent flaws in the interface design. Providing a user-centric and friendly interface will contribute to an increase in operational effectiveness.

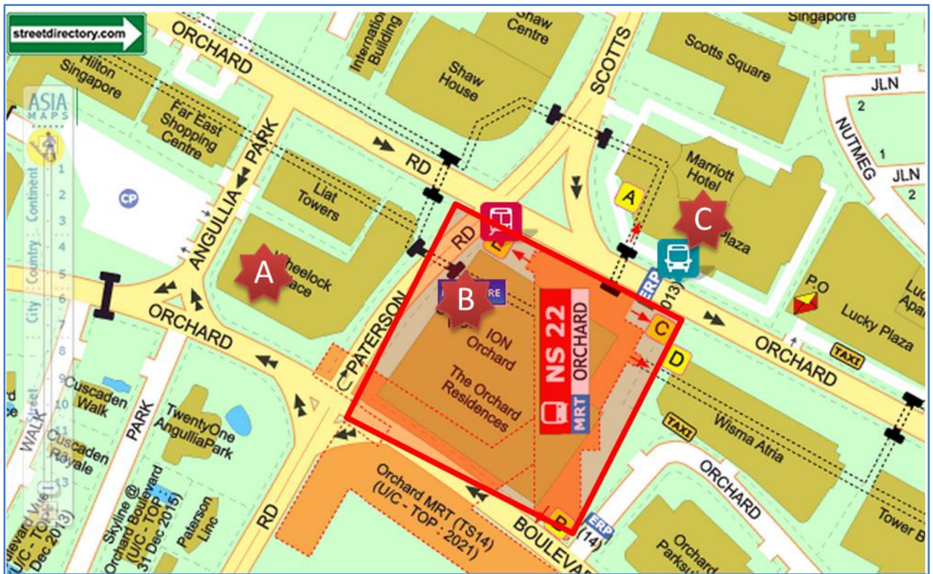
J. THESIS DESIGN SCENARIO

Based on the effective needs identified for the system and operational concept, a design scenario was developed to frame the experimentation portion of this thesis. The scenario takes place in a busy urban shopping belt along Orchard road, Singapore on a typical busy weekend. It takes into account a terrorist group with a considerably high degree of technical expertise, in that they are able to construct concealed devices and obtain hard-to-find components for Improvised Explosive Device (IED) and vehicular bombs. The IEDs and several vehicular bombs were detonated in a string of attacks on multiple sites.

Figure 12. Affected areas of scenario: (A) Wheelock Place – Shopping Arcade/Office. (B) Ion Orchard – Private residence/Retail Outlets/Mass Rapid Transit Station. (C) Marriot Hotel – Hotel establishment/Retail Outlets (Images from Google Earth)

Vehicular bombs were detonated along Paterson Road on the side of (B) Ion Orchard (as shown by the label 1 in Figure 8.) and along Orchard Road, just in front of the (C) Marriot Hotel (Label 2 in Figure 8). IEDs were also detonated in the Orchard Mass Rapid Transit (MRT) station, a public transportation station situated in the basement of Orchard Ion. Establishment "A" (Wheelock Place) suffers minor damages to infrastructure caused by the detonation. Establishment "B" (Orchard Ion) is the worst hit and suffers massive structural damage causing a partial collapse of the building. Establishment "C" (Marriot Hotel) suffers moderate damage from the vehicular bomb.

For the purpose of this experiment, the focus of this urban search and rescue operation will be within the bounds of Establishment B (180m by 180m area), highlighted in Figure 9.



Area of operations: 180 x 180m area

Figure 13. Area of operations (From www.streetdirectory.com.sg).

Several unmanned systems have been deployed for this operation. Two of such systems are fitted with ground penetrating radar payload that detects the movement of survivors trapped under the rubble. The payload is sensitive to small body movement and breathing. Other systems perform various roles of rubble removal, signal repeater, surveillance and logistic support. The key capability of the unmanned system would be their ability to re-optimize its initial planned trajectory for obstacle avoidance in real-time.

In the next few chapters of this thesis, this design scenario would be used as the basis for setting up experiments involving the regeneration of trajectories where the UAV systems encounter obstacles and are required to maneuver around the obstacles in order to continue performing its mission.

III. SETUP AND MODELING

Based on the thesis design scenario, and in order to test the ability of the trajectory generation methodology, there is a need to replicate the operating environment and translate it to an environment where testing and experimentation can be carried out. In this chapter, the setup of an indoor laboratory to perform trials and experiments on the feasibility of a trajectory optimization methodology is described. The trajectory generation methodology and the experiments performed are detailed in Chapter IV and V respectively.

A. LABORATORY SETUP

1. Autonomous Systems Engineering and Integration Laboratory

All trials and experiments were conducted in the Autonomous Systems Engineering and Integration Laboratory (ASEIL) within Bullard Hall of the Naval Postgraduate School (NPS). ASEIL is designed so that all experiments can be performed in a controlled and safe environment. Additionally, the laboratory space can be reconfigured to adapt to the needs of the researcher. The equipment available in the ASEIL consists of multiple Qball-X4 quadrotors, an indoor localization system (Optitrack) and a ground control station. Figure 14 presents the overview of the laboratory layout.

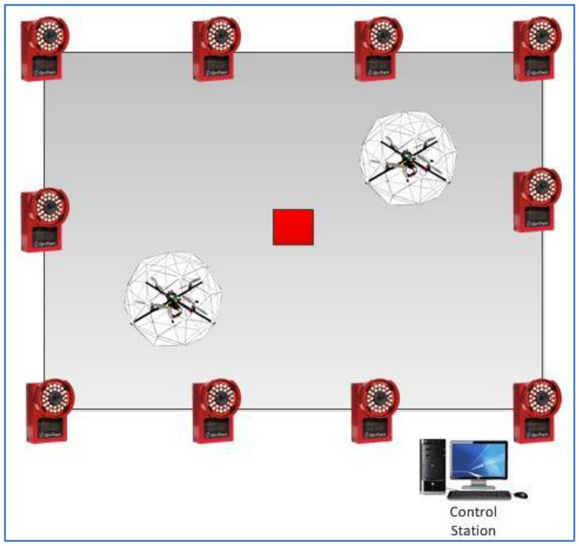


Figure 14. Laboratory setup overview

The floor of the laboratory is covered by reconfigurable, interlocking rubber mats that are used to reduce glare and reflections that might cause interference with the Optitrack system. This rubber foundation also provides a cushioning effect for the quadrotors in the event of a system failure. The ground station in the laboratory setup is a Windows 7 personal computer with 3.20 Ghz processor and 12 GB of RAM. It is positioned on the edge of the room and in front of the operating space for the vehicles. All lab components can be operated via this ground control station.

Matlab/Simulink is the primary software used during all trials. The lab makes use of QuaRC real-time control software as well as the OptiTrack Tracking Tools package that manages the OptiTrack camera system. Both programs are fully integrated with Simulink to capture and track the quadrotor motion. Controlling the vehicles involves running Simulink models on the ground station computer which transmits all data to the vehicles using an adhoc wireless network. Of the Simulink models, the host model gathers data from the OptiTrack system as well as the USB joystick for motion control. The control models, which are linked to each specific vehicle, compile and download code to the Gumstix processors on board the quadrotor. In the next section, the hardware used in the laboratory is described in detail.

2. Hardware

a. Quadrotor

The primary platform utilized in the experiment is Quanser's Qball-X4 quadrotor UAV as shown in Figure 15.

Figure 15. Quanser Qball-X4 Quadrotor UAV

The Quanser Qball-X4 is a rotary wing vehicle platform suitable for a wide variety of UAV research applications (Quanser 2009). It is a quadrotor helicopter design with four motors and speed controllers fitted with 10-inch propellers. Being and open-architecture UAV, the Qball-X4 allows researchers to swiftly create and deploy unmanned vehicle controllers ranging from low-level flight dynamics stabilization to advanced multi-agent guidance, navigation and control algorithms. The specifications of the Qball-X4 include (Quanser 2009):

- Quanser HiQ Data Acquisition Card
- Diameter 0.7m
- 2 LiPo batteries, 2500mAh, 3-cell
- 15 minute flight time per charge
- 4 (740Kv) motors fitted with 10 inch propellers

- Protective carbon fiber cage enclosing motors, propellers and embedded computer
- Wireless communications
- Qball-X4 weight with batteries: 1410g
- Maximum payload capability: 400g

(1) Frame and Protective Cage. The entire mechanism is enclosed within a flexible carbon fiber cage that serves primarily as a form of protective cage. This design ensures safe operation as well as opens the possibilities for a variety of novel applications. As the system is designed for indoor usage, the protective frame is a crucial feature where it protects both the system and its users from damage. The frame gives the Qball-X4 an advantage over other existing vehicles that would suffer significant damage if contact occurs between the vehicles and physical obstacles.

(2) HiQ Data Acquisition Card (DAC)/ Gumstix Computer. The HiQ data acquisition card (Figure 16) is integrated with the Gumstix QuaRC target embedded computer. Together, the HiQ and Gumstix provide a powerful tool that can be used to measure vehicle behavior and run QuaRC generated models.

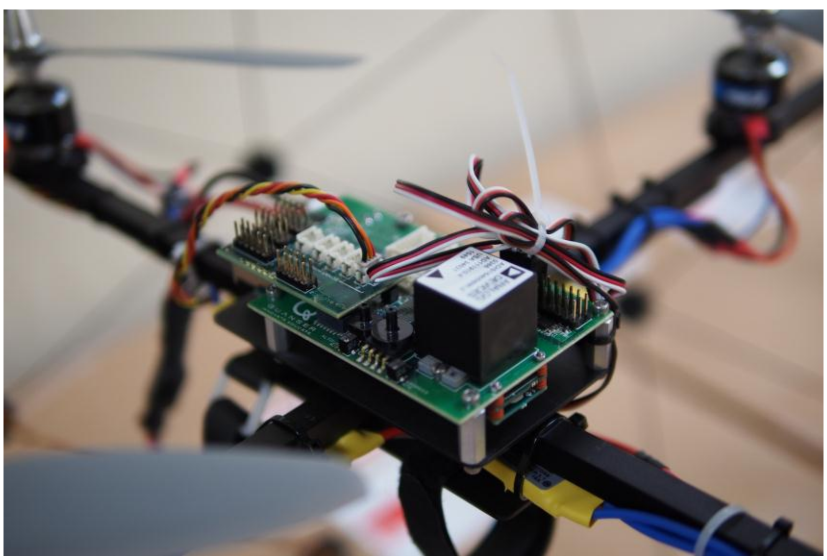


Figure 16. HiQ Data Acquisition Card (DAC)/ Gumstix Computer

Essentially, the DAC is a high-resolution inertial measurement unit and avionics input/output data acquisition card cooperating with the Gumstix embedded computer. Together they control the vehicle by having inputs from the sensors on board and sending motor commands. For the motor speed controllers to be operated, each controller is connected in a specific order to one of the ten PWM servo output channels that are available on the HiQ. An optional daughterboard that contains additional I/O such as receiver or sonar inputs or a TTL serial input used for a GPS receiver. In our laboratory configuration, the sonar daughter board is connected to a sonar device. During operations, HiQ receives information from sensor and motors and performs read/write operations to the vehicle hardware. On the other hand, the Gumstix Verdex is a small form-factor, lightweight embedded computer that runs a Linux-based operating system with wireless communication capability. Configured as a QuaRC target, the Gumstix Verdex enables QuaRC models to be seamlessly downloaded and executed remotely from a host computer. The advantage of this host-target structure is the ease and speed with which operators can build and run their mission controllers. HiQ data acquisition board consists of the following input/output items (Quanser 2009):

- 10 PWM outputs (servo motor outputs)
- 3-axis gyroscope, range configurable for ±75°/s, ±150°/s, or ±300°/s, resolution
- 01832°/s/LSB at a range setting of ±75°/s
- 3-axis accelerometer, resolution 2.522 mg/LSB
- 10 analog inputs, 12-bit, +3.3V
- 3-axis magnetometer, 0.76923 mGa/LSB
- 4 Maxbotix sonar inputs, 1 inch resolution
- Serial GPS input
- 8 channel RF receiver input
- USB input for on-board camera (up to 9fps)
- 2 pressure sensors, absolute and relative pressure
- Input power 10-20V

(3) Sensors and Communication. The Qball is equipped with several sensors. However, not all of these were used in the control of the vehicle. For instance, the magnetometer, which has a listed accuracy of 0.5 mGa/LSB, proved to be unreliable in the indoor laboratory environment. The inconsistency in this sensor's measurements is most likely due to the large amount of unshielded wiring and the construction of the building. As a result, the gyroscope and accelerometer are the sensors chosen to control the roll, pitch, and yaw models of the vehicle. With regard to height control, the sonar altimeter is chosen because it provides very consistent measurements. The sonar used in this experiment is the Maxbotix XL-Maxsonar EZ3 (Figure 17).

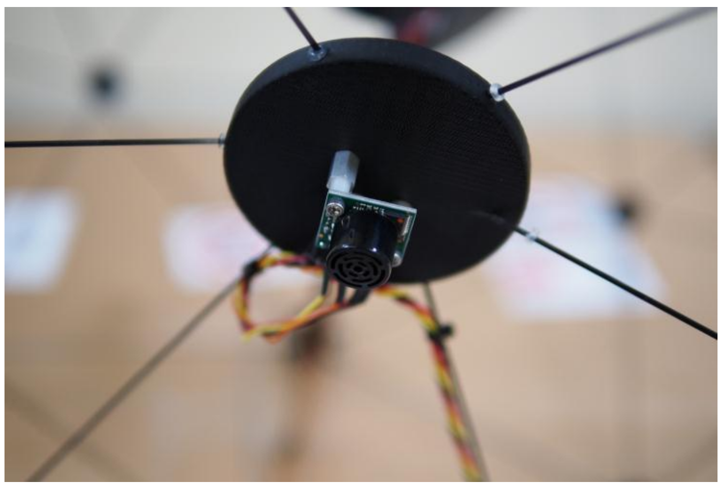


Figure 17. Maxbotix XL-Maxsonar EZ3

This sonar takes readings at a 10 Hz rate and draws very little current. It has a range of 20–765 centimeters and a resolution of 1 cm (Quanser 2011). The sonar is fixed to the bottom of the Qball cage so the vehicle pitch and roll must be accounted for in the height control model of the vehicle. Also, a correction must be made to account for the height difference between where the sonar is located and the center of the body-fixed coordinate frame.

b. Optitrack Tracking System

The Optitrack tracking systems made by Natural Point is an inexpensive indoor camera-based localization and tracking system. The Qball-X4 is able to achieve

localization by means of the external OptiTrack motion capture system. The system uses light emitting diodes and infrared cameras to track the position of passive optical markers placed on the vehicle. Figure 18 shows one of the many IR cameras in an Optitrack camera system setup. Each camera has a field of view of 46 degrees and a resolution of 640x480 pixels at a frame rate of 100 frames-per-second.

Figure 18. Natural Point's Optitrack Tracking Camera

In the absence of a GPS signal, OptiTrack is utilized to track vehicle position and orientation. It allows users the ability to track multiple reflective markers simultaneously in the 3-D space. Figure 19 presents the reflective markers affixed to the quadrotor. When more than one quadrotor is utilized, it is vital that the reflective markers on each system are placed in a unique arrangement. This is to facilitate the tracking of each system.

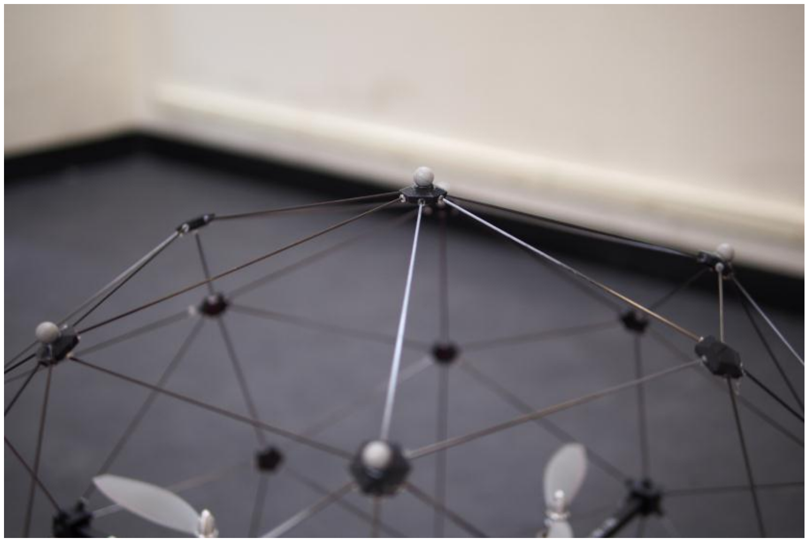


Figure 19. Reflective markets for the quadrotors

In this laboratory, ten stationary Optitrack cameras are laid out, and the system has the capability of tracking up to 32 rigid bodies. The IR cameras are mounted along the ceiling in a manner to eliminate any blind spots in the camera capture volume. The capture volume is the region where the OptiTrack system can successfully track a passive marker. The cubes in Figure 20 represent the approximate capture volume in the lab setup used for this thesis. The pyramids represent the cameras. The capture volume for this lab is approximately 10 feet tall with a width of 12 feet and length of 18 feet.

The exact coordinates of the camera positions can be extracted from the Tracking Tools software (described in Section 3b of this chapter). However, this information is not easily retrievable. In order to extract the information, a simple C++ program (refer to Appendix A) was written. The program uses Tracking Tools's application programming interface (API), a set of C/C++ function calls and a loadable dynamic-link library (DLL) to extract the location information. The exact locations of the cameras are presented in Table 3. The cameras were placed to optimize the horizontal delusion of precision.

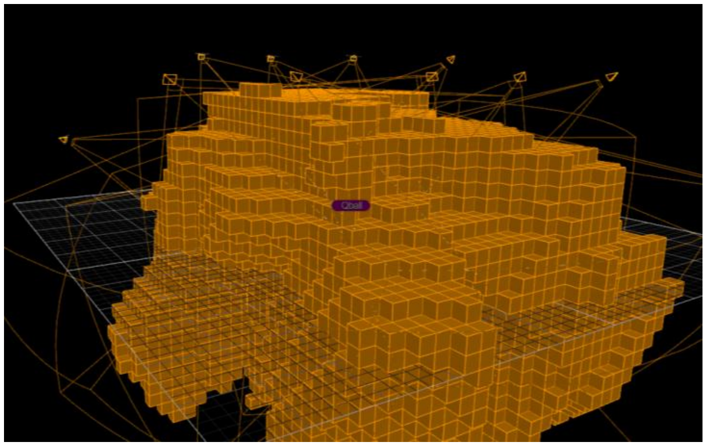


Figure 20. Approximate volume capture in the laboratory setup

Table 3. Location of Optitrack cameras

	Location of Cameras (Optitrack Coordinate System)			
Camera Number	X (m)	Y (m)	Z (m)	ID#
1	3.531	3.376	3.819	151792
2	5.399	3.369	4.390	148184
3	1.860	3.347	4.413	148188
4	-1.740	3.344	4.445	148186
5	0.247	3.349	4.425	148187
6	-2.028	3.620	0.718	151790
7	-2.125	3.677	-2.801	148183
8	0.327	3.659	-2.785	148185
9	2.113	3.692	-2.788	151791
10	4.694	3.636	-2.779	151789

3. Software

a. Quanser QuaRC Toolbox

Quanser Real-time control (QuaRC) is a control software suite (Quanser 2009) that is used by users for both teaching, and research and development applications. It is both compatible and integrated with Simulink and Real Time Workshop. Without the need for a researcher’s effort in hand coding, QuaRC allows the user to graphically draw a controller, generate code and run the experiment. The Simulink-designed controller can be converted into real-time code allowing it to be operating on many target processor and

operating systems combinations. In order to allow for rapid design, test iterations and diagnostics with little to no recompilation required, the control parameters are designed to allow variation while the code is running. The suite also allows for running of multiple controllers on the same processor simultaneously. This can also be achieved even if the controllers are distributed across many processors controlled independently, or even if they are on different chipsets running different operating systems.

QuaRC blocks/tools are relatively extensible and can be scaled to include more systems and commands. This can allow Simulink model to communicate with third party devices, while it provides the mathematical framework for controlling the various devices. The most important ones are:

- Communication blocks for the supported communication protocols, like TCP/IP, UDP, Serial port and Shared memory.
- Hardware-In-the-Loop (HIL) block set, an extensible hardware in-the-loop API used to interface with over 50 data acquisition cards.
- The Vehicle Abstraction Layer (VAL) block set, consisting of a series of blocks that provide a group of high-level primitive commands to the operator, while the VAL deals with the vehicle hardware communication.
- Through the use of the Virtual Reality Toolbox in Simulink and QuaRC, an interactive 3D environment with haptic feedback can be created, allows for simulation or training applications.

b. Optitrack Tracking Tools

The OptiTrack Tracking Tools software package is fully integrated with Simulink and the QuaRC toolbox. The QuaRC OptiTrack block set provides the user with the capability of tracking numerous passive optical markers simultaneously in 3-D environment (NaturalPoint Corporation 2011). One of the advantages of the OptiTrack system is that calibration can be performed relatively quickly.

(1) Calibration

The calibration process is simple and involves the use of two tools; a trident with passive optical markers on the tips and an L-shaped tool that is used to mark the zero point of the room (Figure 21).

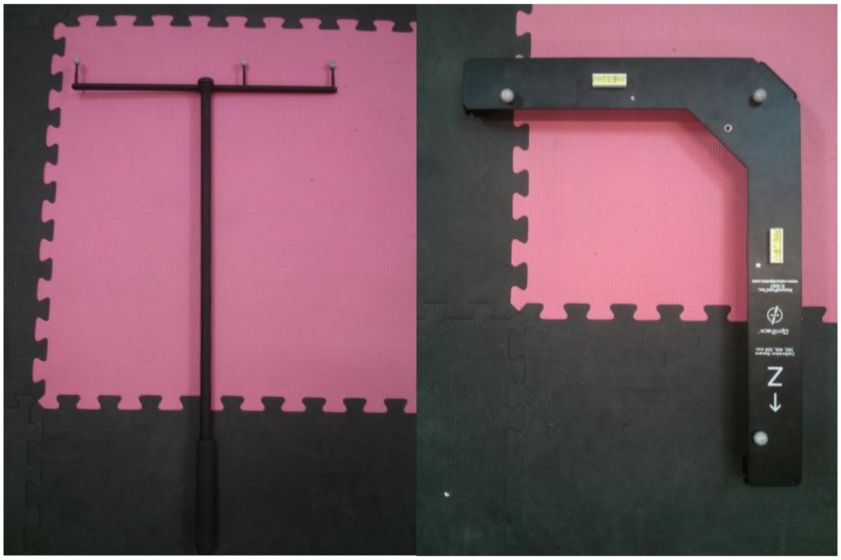


Figure 21. a) Trident, b) L-shape tool

The process involves first performing a visual check of each camera view to ensure there are no false reflections from objects in the camera field of view. If the reflecting object is not easily removable from the workspace, then the software allows one to place a virtual mask over this reflection. For instance, referring to Figure 22, camera 4 and camera 6 appears in each other field of view due to their placement. A virtual mask is applied to mask over their presence.

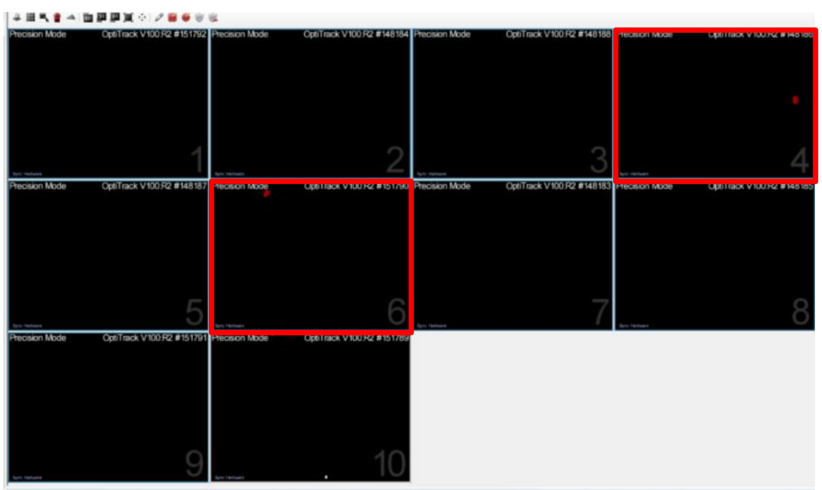


Figure 22. Masking of visible reflections

After this check is complete, the user begins "wanding" by moving the trident in steady and methodological figure-of-eight pattern throughout the camera's field of view. During the process, the software provides feedback of the general quality of the calibration in the form of a color-coded box (Figure 23). Green indicates that the required quality of the calibration can be achieved with the data collected.

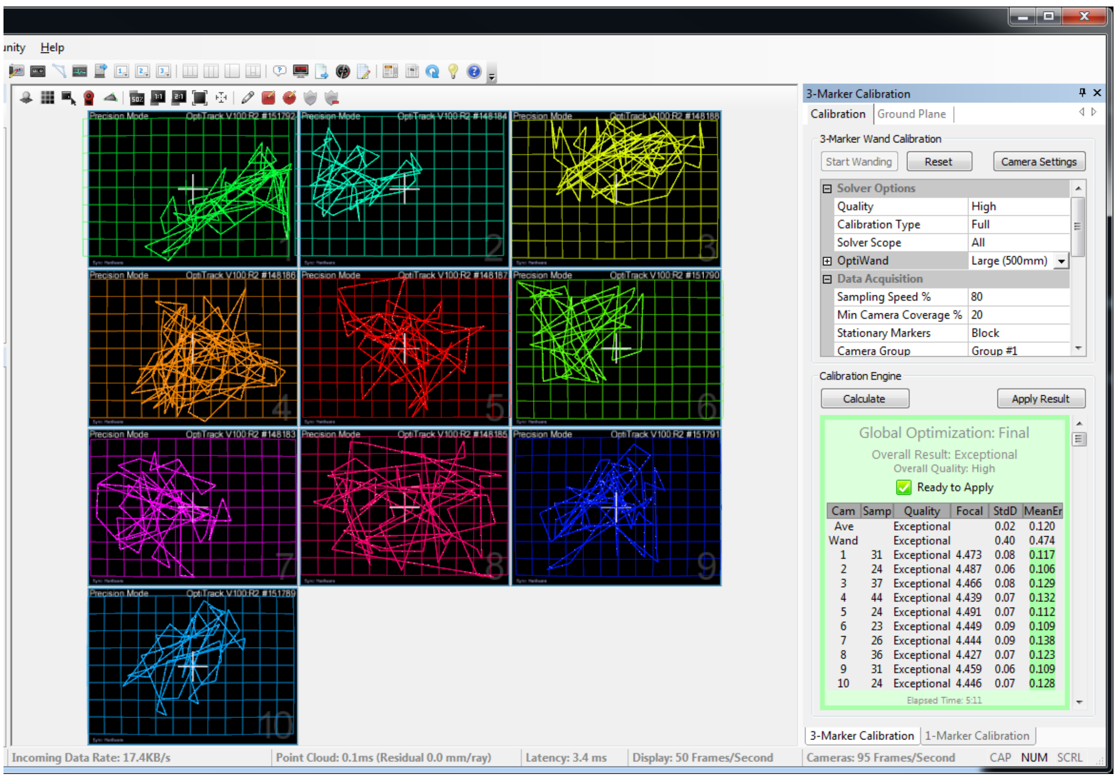


Figure 23. Optitrack camera calibration

When the desired quality is achieved ("exceptional" in this case), the user will initiate calculation on the program. The final step of the calibration involves the placement of the L-shaped ground plane tool in the envisaged center of origin. The software then sets this as the origin from which all measurements will be based. The calibrated camera profile is then saved. All experiments will make use of the same calibration file unless the positions of the Optitrack cameras are changed.

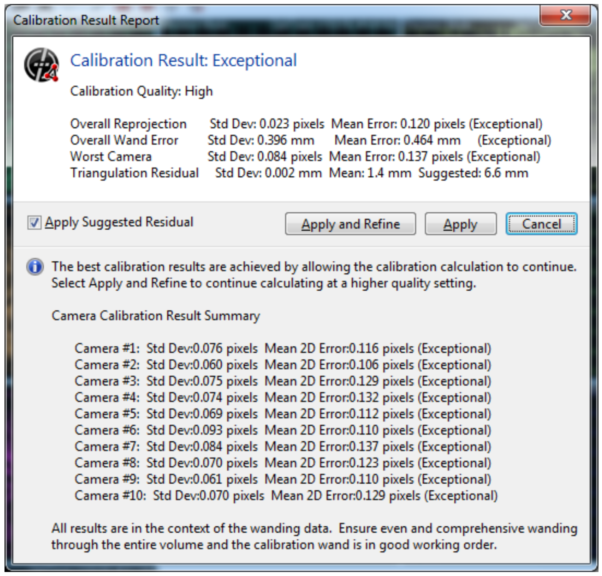


Figure 24. Calibration results

(2) Defining systems to be tracked

Another important feature of the Tracking Tools Software is the ability to create “trackables.” Trackables built upon the original calibration file and add unique identifiers to the systems to be used in the experiment. In order to allow the tracking tools to differentiate between each system, the systems are identified by the unique arrangements of passive optical markers fixed to them. The systems x-y-z position and their angular orientation can be tracked by the tracking tools suite. It is important to make adjustments to the actual height of the quadrotor. As the markers are generally placed on the top of the system, the height registered by the Optitrack system is artificially higher. Using the Optitrack Tracking Tools software, this error can be corrected. Next, the coordinate systems that are used in this thesis will be discussed in the following section.

B. MODELING

Prior to performing any obstacle avoidance or trajectory generation, the motion and control of the vehicles must be understood. Feasible collision-avoidance trajectories require knowledge of the physical parameters and correct control inputs for the vehicle. In this section, the simplifying assumptions about the operating environment and vehicles

will be outlined. This will be followed by a section about coordinate frame designation and sections about the control and modeling of the Qball-X4. The modeling of the vehicle is outlined according to state space format. This means that the dynamics of each vehicle is contained in a set of differential equations which is represented in matrix format.

1. Assumptions

Several justifiable assumptions are made to simplify the modeling of the complex dynamics of the quadrotor:

- The Earth is flat and not rotating.
- Constant acceleration of 9.81 m/s^2 due to gravity
- The quadrotor is a rigid body that does not flex.
- Drag forces are assumed to be negligible (Speed of the experiment are kept low).
- Pitch and roll angles of the Quadrotor throughout the flight are small.

2. Coordinate Frames

Two main coordinate frames will be used in this paper, namely the body-fixed frame and the Optitrack- coordinate frame. Figure 25 presents the body-fixed coordinate frame of the quadrotor. The frame of this coordinate system is attached to the center of mass of the quadrotor and it rotates with the vehicle. From the same figure, the rear of the vehicle is marked by an orange marking tape. The X axis aligns with direction of forward movement, the Y axis points to the left, and the Z axis points up.

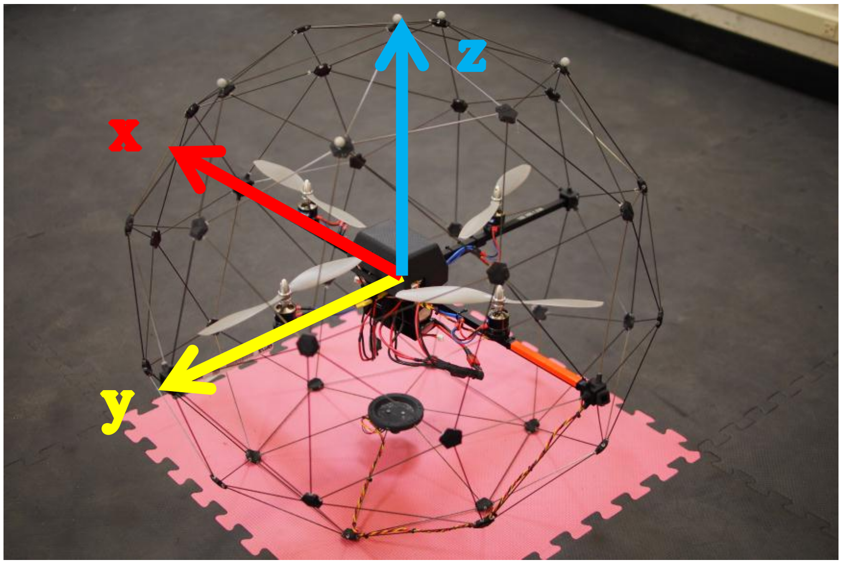


Figure 25.   Body-fixed coordinate frame

The roll of the vehicle (x-axis) is along the axis of the two opposing propellers and point toward the front of the vehicle. The pitch occurs on the y axis which points to the left of the vehicle, while the yaw occurs on the z axis that points upwards of the vehicle. The right hand rule is applied to determine directions of the various Euler angles for the vehicle. A positive roll direction is counterclockwise about the x axis when facing the quadrotor. This rule applies for pitch and yaw direction. This coordinate system is used to define the dynamics model of the quadrotor. The physical model of the quadrotor will be presented in the next section.

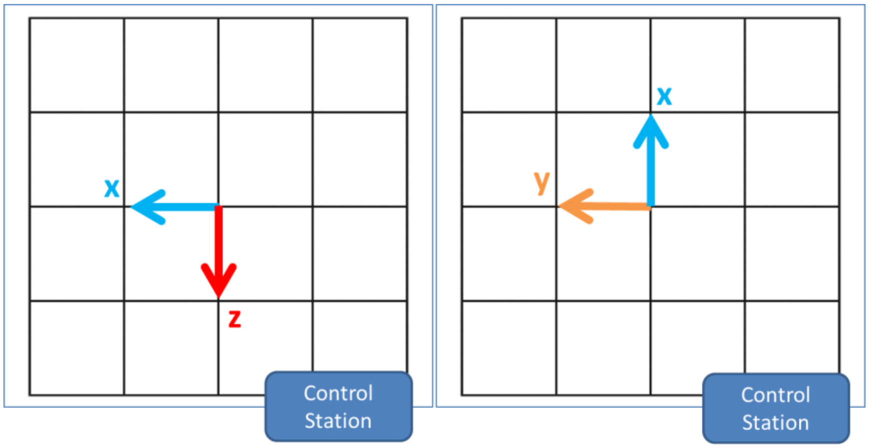


Figure 26.   a) Optitrack coordinate system (left), b) Body-fixed coordinate system (right)

The second coordinate system, the Optitrack coordinate system (Figure 26b), is utilized as the positioning tracking system for the UAVs. It serves to approximate the Earth as a flat, non-moving object. This approximation is possible because the Qball(s) are not operating at fast speeds or traveling long distances. Thus, the effects of the Earth's curvature and sidereal motion can be ignored. The frame itself is fixed on the ground at the center of the indoor lab. The x axis points toward the left of the lab from the control station and z axis toward the control station. Y axis is pointing upwards from the ground. It is important to note this difference as the trajectory generated by the direct method algorithm is in a different coordinate system and needs to be accounted for when applying it to the flight model.

3. Qball-X4 Physical Modeling

The quadrotor is propelled by four symmetrically-pitched and independently controlled rotor blades. In general, the movement of the system is controlled by altering either the pitch of the blades, or their rotational rates. For the Qball-X4 quadrotor, no complicated pitch control mechanism exists. Therefore, its motion is fully dependent on the variation of thrust on the individually controlled fan blades. Some of the system parameter as determined by Quanser is tabled in Table 4.

Table 4. Quanser Qball-X4 system parameter (Quanser 2011)

Parameter	Value
K	120 N
ω	15 rad/s
J	0.03 kg m^2
M	1.4 kg
K_y	4 N m
J_{yaw}	0.04 kg m^2
L	0.2 m

Referring to "Direct Method Based Control System for an AutonomousQuadrotor" (Yakimenko et al. November 2010, 285–316), a simplified model of the quadrotor's dynamic can be obtained by considering the following. The

schematic of the quadrotor is presented in Figure 27, showing the numberings of the motor, as well as the convention of the various parameters.

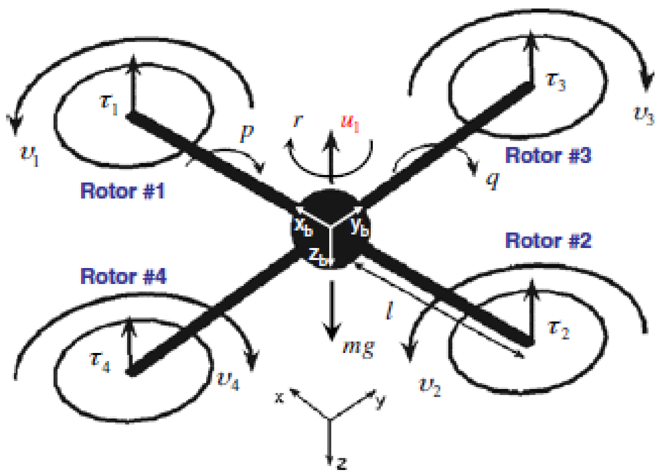


Figure 27. Quadrotor Schematic (From Yakimenko et al. November 2010, 285–316).

Let v_i and τ_i be the normalized torque, and normalized thrust for the *i*th rotor, respectively, where $i = 1,\ldots,4$. The distance of the rotor from the center of mass of the quadrotor is defined as l. Similarly, let $\tilde{u}_i$ be the set of four controls in the body frame, as functions of normalized individual thrusts and torques. The total normalized thrust in the body frame $\tilde{u}_1$ is given by:

$$\tilde{u}_1 = \left(\tau_1 + \tau_2 + \tau_3 + \tau_4\right); \tag{1}$$

Subsequently, the roll moment can be achieved by varying the left and right rotor speeds:

$$\tilde{u}_2 = l\left(\tau_4 - \tau_3\right); \tag{2}$$

The pitch moment can be generated by varying the front and back rotor speeds:

$$\tilde{u}_3 = l\left(\tau_1 - \tau_2\right); \tag{3}$$

The yaw moment can be obtained from the difference in the counterclockwise and clockwise normalized torques of each rotor:

$$\tilde{u}_4 = \left(v_3 + v_4 - v_1 - v_2\right) \tag{4}$$

Further, an introduction of a twelve-state vector of

$$x = \left[x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi} \right]^T \tag{5}$$

where $[x, y, z]^T$ represents the translational position of the quadrotor center of gravity in the NED frame, $[\phi, \theta, \psi]^T$ is the attitude vector where ϕ, θ, ψ are the roll, pitch, and yaw angle respectively between $[x, y, z]$ and the body frame.

In (Castelli 2012), it is seen that the translational equations of motion are derived using rotational matrix $R_{zyx}(R_{\psi\theta\phi})$, instead of the conventional $R_{xyz}(R_{\phi\theta\psi})$. These equations of motion are given by:

$$\ddot{x} = -u_1 \cos\phi \sin\theta \tag{6}$$

$$\ddot{y} = u_1 \sin\phi \tag{7}$$

$$\ddot{z} = g - u_1 \cos\phi \cos\theta \tag{8}$$

The desired outputs of the system are the translational position and the yaw angle. Thus, by defining the control vector **u** from the total normalized thrust and second derivatives of the Euler angles, and developing the equations of motion by utilizing the rotational matrix, the complete set of equations for the state vector (Hargraves and Paris 1987, 338–342) is derived as follows:

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ -\cos\phi\sin\theta u_1 \\ \sin\phi u_1 \\ g-\cos\phi\cos\theta u_1 \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{9}$$

To identify the relations between the three controls (u_2, u_3, u_4) defined in the local tangent, and those defined in the body frame ($\tilde{u}_2, \tilde{u}_3, \tilde{u}_4$), a relationship between the body rates and the Euler rates can is first determined.

$$\begin{aligned} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \psi \end{bmatrix} = &\begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \\ &+\begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \end{aligned} \tag{10}$$

Thereafter, by assuming small rates, and by differentiating(10), the relationship of the controls in the body frame is found as follows:

$$\begin{bmatrix}\tilde{u}_2\\ \tilde{u}_3\\ \tilde{u}_4\end{bmatrix} \approx \begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}\cos\psi & \sin\psi\cos\phi & 0\\ -\sin\psi & \cos\psi\cos\phi & 0\\ 0 & -\sin\phi & 1\end{bmatrix}\begin{bmatrix}u_2\\ u_3\\ u_4\end{bmatrix} + \begin{bmatrix}-\sin\psi\dot{\psi} & \cos\phi\cos\psi\dot{\psi} & 0\\ -\cos\psi\dot{\psi} & -\cos\psi\sin\phi\dot{\phi} & 0\\ 0 & -\cos\phi\dot{\phi} & 0\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \tag{11}$$

In the next chapter, the control methodology of the quadrotor will be discussed and presented in details.

IV. CONTROL AND OPTIMIZATION METHODOLOGY

A. INTRODUCTION

The optimal control problem this thesis addresses is to guide an aerial system from an initial state to some final state with constraints imposed on both the states and the controls. Ideally, the routine that is used should be capable of updating itself multiple times over the course of the trajectory to mitigate disturbances and un-modeled motor dynamics.

The concept behind direct methods is the use of a finite set of variables to arrive at an optimal or quasi-optimal solution. This approach involves using a function to approximate the states and controls and a function to represent the cost of the process. Cost here refers to the penalties calculated for each solution, so that an optimal solution can be determined. To further define the methodology used in this thesis, it uses the direct method of calculus of variations exploiting the inverse dynamics of a vehicle in the virtual domain (IDVD). The key feature of the IDVD method is that it uses inverse dynamics where the state and control inputs can be expressed as functions of the output and its derivatives can be termed 'differentially-flat." As a result, all optimization occurs in the output space as opposed to the control space.

Another feature of this method is the application of optimization in the virtual domain, as opposed to the time domain, decoupling time and space parameterization. This lifts the restrictions faced by the main stream direct methods and allows for fast prototyping of optimal trajectories (Yakimenko et al. November 2010, 285–316). Since the IDVD method uses significantly lesser parameters, the computational requirement of using this methodology is significantly lesser than that of other direct methods. The advantages of the IDVD method allows for a real-time quasi-optimal mission generation capability for the systems involved. As the method is relatively easy to modify and code, it results in increased flexibility (with regard to mission scenarios) for the operator.

In this chapter, the controller architecture for controlling the unmanned system is first presented. This is followed by a detailed discussion of the trajectory optimization process using the IDVD methodology.

B. CONTROLLER ARCHITECTURE

The architecture of the proposed controller is presented in Figure 28. As shown in the figure, this proposed architecture is made up of two distinct entities, the trajectory follower (upper portion) and the trajectory generator (lower portion).

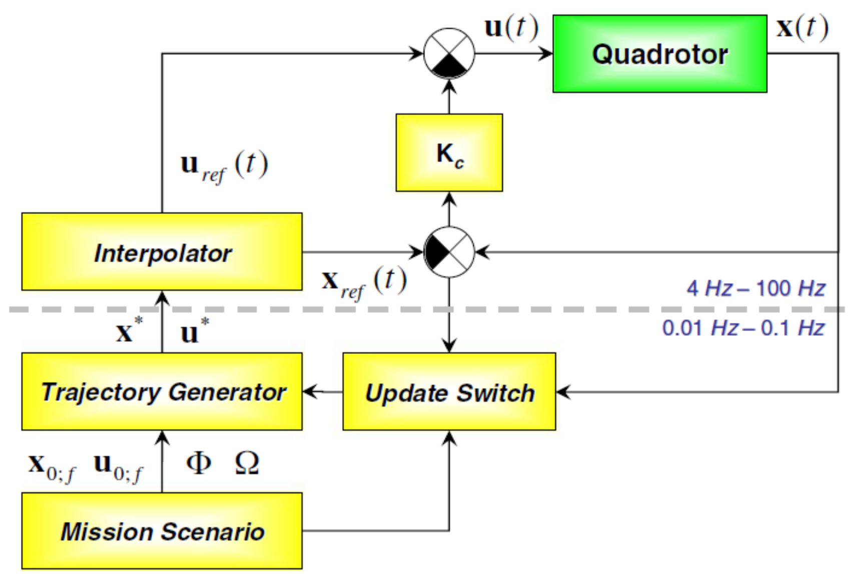


Figure 28. Proposed architecture of controller (From Yakimenko et al. November 2010, 285–316).

Depending on the mission objectives and operating scenario, the trajectory generator of this controller will produced a plausible quasi-optimal path for the system. This trajectory produced is the best possible route calculated based on the initial and final conditions. However, the resulting route might not be the most optimal route, but it is definitely a feasible route that is calculated in a relatively short time. As mentioned earlier, this quasi-optimal route obtained in a shorter timeframe is deemed to be a good trade-off against an optimal route that takes a long time to compute. This ability of the trajectory generator to generate routes in real time allows for the re-optimization of

trajectory during flight. Due to the dynamic nature of an urban search and rescue environment, the initially cleared pathway may be obstructed by new obstacles. The ability that allows for the regeneration of trajectory in flight will be invaluable to operators.

During the mission, the objectives for the mission may change or the discrepancy between the current state and the suggested path could become too large. This is possible due to the disturbances and inherent imperfections of the controller. When such an event occurs, the update switch forces the trajectory generator to calculate a new quasi-optimal trajectory passing through the current state to count it as the new vector of initial conditions. Depending on the mission and the on-board computational power, it is possible that the trajectory generator would update the reference trajectory periodically every 10 to 100 seconds.

The other vital component of this controller is the inner loop that uses a linear quadratic regulator (LQR) to track and monitor the trajectory of the system in the presence of disturbances (Cowling, Whidborne, and Cooke 2006). In essence, since the interpolator produces updates at a high frequency rate (4–100 Hz), the controller can perpetually track the reference trajectory to ensure that the LQR controllers counters any disturbances experienced by the system in a timely fashion.

C. INVERSE DYNAMICS IN THE VIRTUAL DOMAIN

IDVD can be described completely via four areas; Generation of a reference trajectory that is independent of time derivative constraints; using the speed function to convert the reference trajectory back into time domain; utilizing inverse dynamics to obtain vehicular control states; and optimization of trajectories by satisfying boundary conditions determined by the user. The following sections describe the application of IDVD method for this thesis.

1. Reference Trajectory

By employing a virtual variable “τ” as the independent variable in parameterizations the method creates a reference trajectory which is independent of any

time derivative constraints. This variable varies between 0 and some finite value τ_f, where τ_f is considered as one of the varied parameters of the trajectory optimization problem. Since τ is a reference function in the virtual domain, the method decouples space and time, allowing an easier time in obtaining the quasi-optimal trajectory.

The IDVD method makes use of different parameterizations approximating three Cartesian coordinates of a moving object to achieve the desired trajectory for the operation(s). In considering the general shape of an expected trajectory, the order of parameterization is determined by the number of initial and final conditions that need to be satisfied. The order of the reference function polynomial, N, is determined by the number of boundary conditions that must be satisfied, and it can be increased to add to the degree of freedom. This is given by the relationship:

$$N = d_0 + d_f + 1 \tag{12}$$

Where d_0 is the highest-order spatial derivative of the initial conditions, and d_f is the highest order of derivatives in the final conditions. In other words, to satisfy up to the second-order derivative constraints both the initial and final portion of the trajectory, a 5th-order parameterization is the least number of parameters required.

The quadrotor's maneuver trajectory can be modeled by a trajectory parameterization utilizing trigonometric terms. The three coordinates (x,y and z) can be represented in the form:

$$\begin{aligned}
x(\bar{\tau}) &= P_x(\bar{\tau}) = a_{x0} + \sum_{i=1}^{3} a_{xi}\cos(i\pi\bar{\tau}) + \sum_{i=1}^{4} b_{xi}\sin(i\pi\bar{\tau}) \\
y(\bar{\tau}) &= P_y(\bar{\tau}) = a_{y0} + \sum_{i=1}^{3} a_{yi}\cos(i\pi\bar{\tau}) + \sum_{i=1}^{4} b_{yi}\sin(i\pi\bar{\tau}) \\
z(\bar{\tau}) &= P_z(\bar{\tau}) = a_{z0} + \sum_{i=1}^{3} a_{zi}\cos(i\pi\bar{\tau}) + \sum_{i=1}^{4} b_{zi}\sin(i\pi\bar{\tau})
\end{aligned} \tag{13}$$

Where $\bar{\tau} = \tau / \tau_f$. N as described by equation (12) is 7, and this is depending on the initial and final boundary conditions. The unknown coefficients in equation (13) can be found by resolving the following matrix of algebraic equations.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pi & 2\pi & 3\pi & 4\pi \\ 0 & 0 & 0 & 0 & -\pi & 2\pi & -3\pi & 4\pi \\ 0 & -\pi^2 & -(2\pi)^2 & -(3\pi)^2 & 0 & 0 & 0 & 0 \\ 0 & \pi^2 & -(2\pi)^2 & (3\pi)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\pi^3 & -(2\pi)^3 & -(3\pi)^3 & -(4\pi)^3 \\ 0 & 0 & 0 & 0 & \pi^3 & -(2\pi)^3 & (3\pi)^3 & -(4\pi)^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \\ x'_0 \tau_f \\ x'_f \tau_f \\ x''_0 \tau_f^2 \\ x''_f \tau_f^2 \\ x'''_0 \tau_f^3 \\ x'''_f \tau_f^3 \end{bmatrix} \quad (13)$$

Equation (14) applies similarly for y and z coordinates. The initial and terminal state (x_0 & x_f), first and second-order derivatives (x'_0, x'_f, x''_0 & x''_f) were considered as constraints to be satisfied and the third-order derivatives (x'''_0, x'''_f) at both ends of the trajectory (the initial and final jerk) are free variables.

2. Speed Factor

Once the trajectory is found in the virtual domain, there is a need to map it back into the time domain. As described in the beginning of this chapter, the argument τ was used to decouple space and time by transforming the function into the virtual domain. When the reference trajectory is generated, there is a need to switch from the virtual domain back into the time domain. This conversion is facilitated by the speed factor (Milionis G. 2011) given as:

$$\lambda(\tau) = \frac{d\tau}{dt} \quad (14)$$

By utilizing this speed factor, the speed profile can be varied along the same trajectory by changing the speed factor:

$$V(\tau) = \lambda(\tau)\sqrt{P_x'^2(\tau) + P_y'^2(\tau) + P_z'^2(\tau)} \quad (15)$$

The initial and final value of λ are set to one (it simply rescales the virtual arc length τ_f) and the 1st order derivative set to zero (for smooth departure and arrival). Then, following the previous discussion, to increase the flexibility of the speed reference profile, the 2nd order derivatives of the speed factor can be used as extra varied

parameters. This requires a 5th-order parameterization. Employing a polynomial of the form of (13) we resolve for the corresponding coefficients utilizing algebraic equations, similar to those of (14):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{6}\tau_f^3 & \frac{1}{12}\tau_f^4 & \frac{1}{20}\tau_f^5 \\ 0 & 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{3}\tau_f^3 & \frac{1}{4}\tau_f^4 \\ 0 & 0 & 1 & \tau_f & \tau_f^2 & \tau_f^3 \end{bmatrix} \begin{bmatrix} a_{\lambda 0} \\ a_{\lambda 1} \\ a_{\lambda 2} \\ a_{\lambda 3} \\ a_{\lambda 4} \\ a_{\lambda 5} \end{bmatrix} = \begin{bmatrix} \lambda_0 \\ \lambda'_0 \\ \lambda''_0 \\ \lambda'''_0 \\ \lambda_f \\ \lambda'_f \\ \lambda''_f \\ \lambda'''_f \end{bmatrix} \tag{16}$$

Thus the speed profile can be computed as:

$$V(\tau) = P_\lambda(\tau)\sqrt{P_x'^2(\tau) + P_y'^2(\tau) + P_z'^2(\tau)} \tag{17}$$

3. Inverse Dynamics

In order to determine the controls for the vehicle, inverse dynamics of the system are needed. By using the differential flatness property of the system, the inverse dynamics of the vehicle can be derived. Differential flatness is the property of a system, such that all of its states and controls can be expressed in terms of the output vector and its derivatives (Koo and Sastry 1999).

The quadrotor used roll and pitch angles in coordination with a yaw angle to control its position in the horizontal plane and the propeller thrust to control its height. The state vector can be expressed as a function of the output vector, and its derivatives as given by:

$$\theta = \arctan\left(\frac{\ddot{x}}{g - \ddot{z}}\right) \tag{18}$$

$$\phi = \arcsin\left(\frac{-\ddot{y}}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + \left(g - \ddot{z}\right)^2}}\right) \tag{19}$$

Taking derivatives of Eq. 23 and 24 the following equations are found:

$$\dot{\theta} = \frac{x\left(g - \ddot{z}\right) + \ddot{x} \cdot z}{\left(g - \ddot{z}\right)^2 + \ddot{x}^2} \tag{20}$$

$$\dot{\phi} = \frac{\left(\ddot{x}\dddot{x} - \left(g - \ddot{z}\right)\dddot{z}\right)\ddot{y} - \left(\ddot{x}^2 + \left(g - \ddot{z}\right)^2\right)\dddot{y}}{\left(\ddot{x}^2 + \ddot{y}^2 + \left(g - \ddot{z}\right)^2\right)\sqrt{\ddot{x}^2 + \left(g - \ddot{z}\right)^2}} \tag{21}$$

The speed factor and the virtual derivatives are then utilized to obtain the time domain derivatives. This results in a trajectory that satisfies the boundary conditions.

4. Cost Function

In the optimization process, one may choose to minimize time, minimize control efforts, or a combination of both parameters to assure a certain attitude for most of the flight. The cost function is a combination of the Performance Index (PI) and includes the weighted penalties, i.e., the sum of the running costs of not meeting the constraints. PI is a quantitative measure of the optimality of the trajectory (Yakimenko et al. November 2010, 285–316). By utilizing the cost function, one can optimize the trajectory to allow the system to perform obstacle avoidance accurately and safety. When operating a single unmanned system, the key constraints are arrival times, vehicle attitude constraints (roll and pitch angles), and obstacle constraints. Due to the limited space available in the ASEIL, the physical dimensions of the flight area were also added as physical constraints. Based on these constraints, the cost function *J* was derived as shown:

$$J = PI + w_1\left(\frac{\left(t_{desired} - t_{end}\right)^2}{t_{desired}^2}\right)$$
$$+w_2\left(\max\left(0,\frac{|\phi|_{\max} - \phi_{threshold}}{\phi_{threshold}}\right)^2 + \max\left(0,\frac{|\theta|_{\max} - \theta_{threshold}}{\theta_{threshold}}\right)^2\right)$$
$$+w_3 \max\left(0,\frac{d_{threshold,Obs} - d_{\min,Obs}}{d_{threshold,Obs}}\right)^2$$
$$+w_4\left(\begin{array}{l}\max\left(0,\dfrac{|X|_{\min} - X_{threshold}}{X_{threshold}}\right)^2 \\ +\max\left(0,\dfrac{|Y|_{\min} - Y_{threshold}}{Y_{threshold}}\right)^2 \\ +\max\left(0,\dfrac{|Z|_{\min} - Z_{threshold}^{\min}}{Z_{threshold}^{\min}}\right)^2 + \max\left(0,\dfrac{|Z|_{\max} - Z_{threshold}^{\max}}{Z_{threshold}^{\max}}\right)^2\end{array}\right) \quad (22)$$

where w_1, w_2, w_3 and w_4 are the weighting factors assignments that are tuned to control each individual penalty. The other parameters are, $t_{desired}$ – desired time of arrival; t_{end} – end time of flight; $\phi_{\max}$ maximum roll of the flight; $\phi_{threshold}$ – roll limit of the controller; θ_{max} – maximum pitch of the flight; $\theta_{threshold}$ – pitch limit of the controller; $d_{threshold,Obs}$ – distance threshold of the obstacle; $d_{min,Obs}$ – allowable distance from the obstacle.

In the case of multiple UAVs, each UAV is supposed to take care of its own control. However, for friendly UAVs, one may also utilize a centralize controller to achieve the desired control. In the latter case, the same cost function augmented with an additional constraint of keeping spatial separation between them, can be used to generate a trajectory for each UAV. The combined cost function J for both UAVs (UAV A and UAV B) is formulated as:

$$
\begin{aligned}
J = PI &+ w_1\left(\frac{\left(t_{desired} - t_{end\,(UAVA)}\right)^2}{t_{desired}^2} + \frac{\left(t_{desired} - t_{end\,(UAVB)}\right)^2}{t_{desired}^2}\right) \\
&+ w_2\left(\begin{aligned} &\max\left(0, \frac{|\phi|_{\max(UAVA)} - \phi_{threshold}}{\phi_{threshold}}\right)^2 + \max\left(0, \frac{|\theta|_{\max(UAVA)} - \theta_{threshold}}{\theta_{threshold}}\right)^2 \\ &+ \max\left(0, \frac{|\phi|_{\max(UAVB)} - \phi_{threshold}}{\phi_{threshold}}\right)^2 + \max\left(0, \frac{|\theta|_{\max(UAVB)} - \theta_{threshold}}{\theta_{threshold}}\right)^2 \end{aligned}\right) \\
&+ w_3 \max\left(0, \frac{d_{threshold,Obs} - d_{\min,Obs}}{d_{threshold,Obs}}\right)^2 \\
&+ w_4\left(\begin{aligned} &\left(\begin{aligned} &\max\left(0, \frac{|X|_{\min} - X_{threshold}}{X_{threshold}}\right)^2 \\ &+ \max\left(0, \frac{|Y|_{\min} - Y_{threshold}}{Y_{threshold}}\right)^2 \\ &+ \max\left(0, \frac{|Z|_{\min} - Z_{threshold}^{\min}}{Z_{threshold}^{\min}}\right)^2 + \max\left(0, \frac{|Z|_{\max} - Z_{threshold}^{\max}}{Z_{threshold}^{\max}}\right)^2 \end{aligned}\right)_{UAVA} \\ &+ \left(\begin{aligned} &\max\left(0, \frac{|X|_{\min} - X_{threshold}}{X_{threshold}}\right)^2 \\ &+ \max\left(0, \frac{|Y|_{\min} - Y_{threshold}}{Y_{threshold}}\right)^2 \\ &+ \max\left(0, \frac{|Z|_{\min} - Z_{threshold}^{\min}}{Z_{threshold}^{\min}}\right)^2 + \max\left(0, \frac{|Z|_{\max} - Z_{threshold}^{\max}}{Z_{threshold}^{\max}}\right)^2 \end{aligned}\right)_{UAVB} \end{aligned}\right) \quad (23) \\
&+ w_5 \max\left(0, \frac{d_{threshold\,(UAVs)} - d_{\min(UAVs)}}{d_{threshold\,(UAVs)}}\right)^2
\end{aligned}
$$

THIS PAGE IS INTENTIONALLY LEFT BLANK

V. RESEARCH SCENARIO AND RESULTS

A. INTRODUCTION

Adapted from the thesis design scenario, specific sub-scenes are conceived and experiments were carried out to test the ability of the trajectory generation ability of the proposed approach. The experimentation part of the thesis serves as validation and verification of the direct method approach (presented in Chapter IV) and its envisaged performance in the cluttered urban search and rescue environment. In this chapter, the approach of the experiments is first depicted, and this is followed by the descriptions of the various simulated scenarios. Finally the results from the ensuing experiments are presented.

B. APPROACH

All trials and experiments were conducted in the ASEIL. It is designed so that all experiments can be performed in a controlled and safe environment. The equipment available in the laboratory consists of multiple Qball-X4 quadrotors, an indoor localization system (Optitrack) and a ground control station. These systems were presented in Chapter III, Section A.

The ground station is positioned on the edge of the room and in front of the operating space for the vehicles. This position gives the user a good situational awareness of the simulation. All lab components can be operated via the ground control station. Matlab/Simulink is the primary software used for the simulations performed. The QuaRC real-time control software and the OptiTrack Tracking Tools package manage the OptiTrack camera system and provide the motion capturing and tracking capabilities required to perform the experiments. Generally, there are two steps to the experiments and they are trajectory generation and mission execution. These are described in the following section.

1. Trajectory Generation

Based on the methodology described in Chapter IVC, the trajectories of the experiments are developed in the Simulink modeling environment. An optimization script is written to perform trajectory generation. The script is presented in Appendix B. A Simulink model (Figure 29) is created that solves equations (13) and (16) and creates the trajectory as a function of time. Within the trajectory, arrival time, all vehicle controls, and/or the relative distance between vehicles can be calculated for every point along the trajectory.

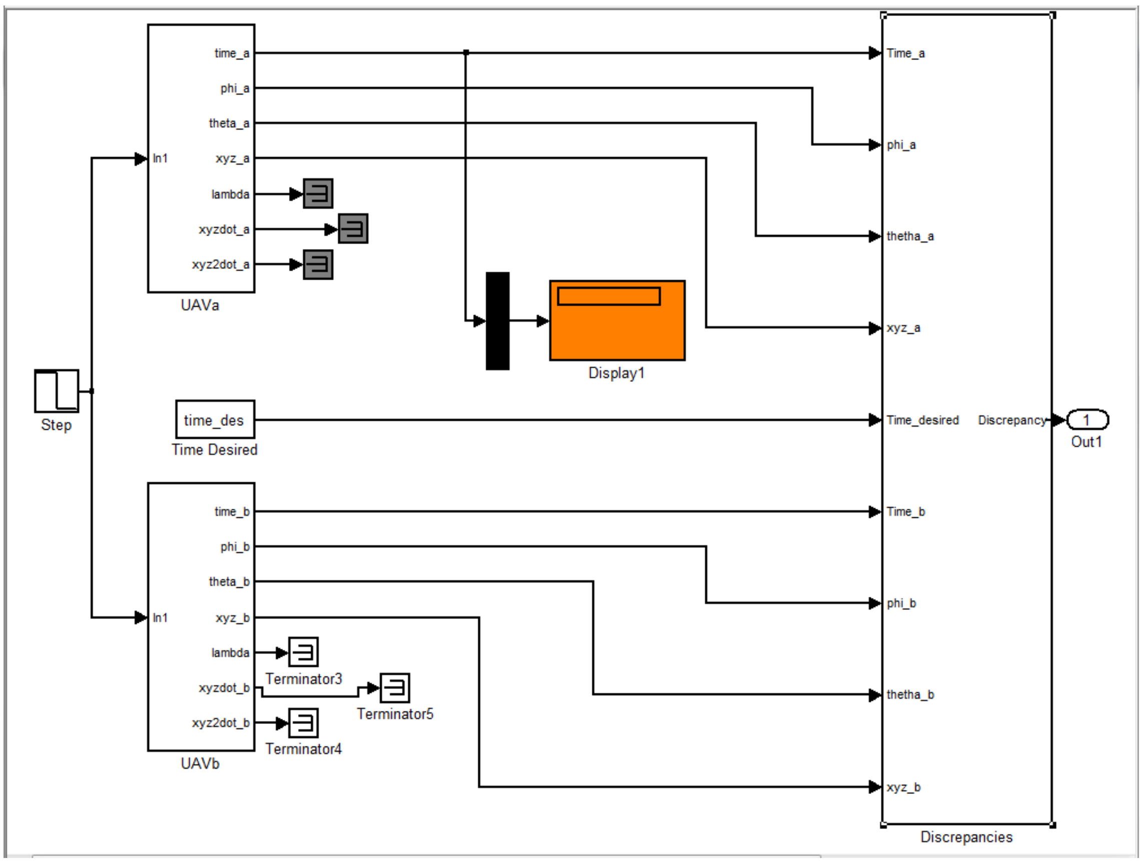


Figure 29. Trajectory generator Simulink model

Having identified the plausible trajectory, the cost function is applied to determine if it is indeed the best. The fminsearch command is used inside Matlab to run the Simulink model and determine a set of varied parameters that minimizes the cost function, hence, a quasi-optimal trajectory. The cost function, as specified by the user, is

implemented by the discrepancies sub-model shown in Figure 29 and elaborated in Figure 30. When the solution is found, a final Simulink model is called to translate the results from the optimization process. The trajectory is now ready to be transferred to the quadrotor for mission execution. The mission execution process is described in the next section.

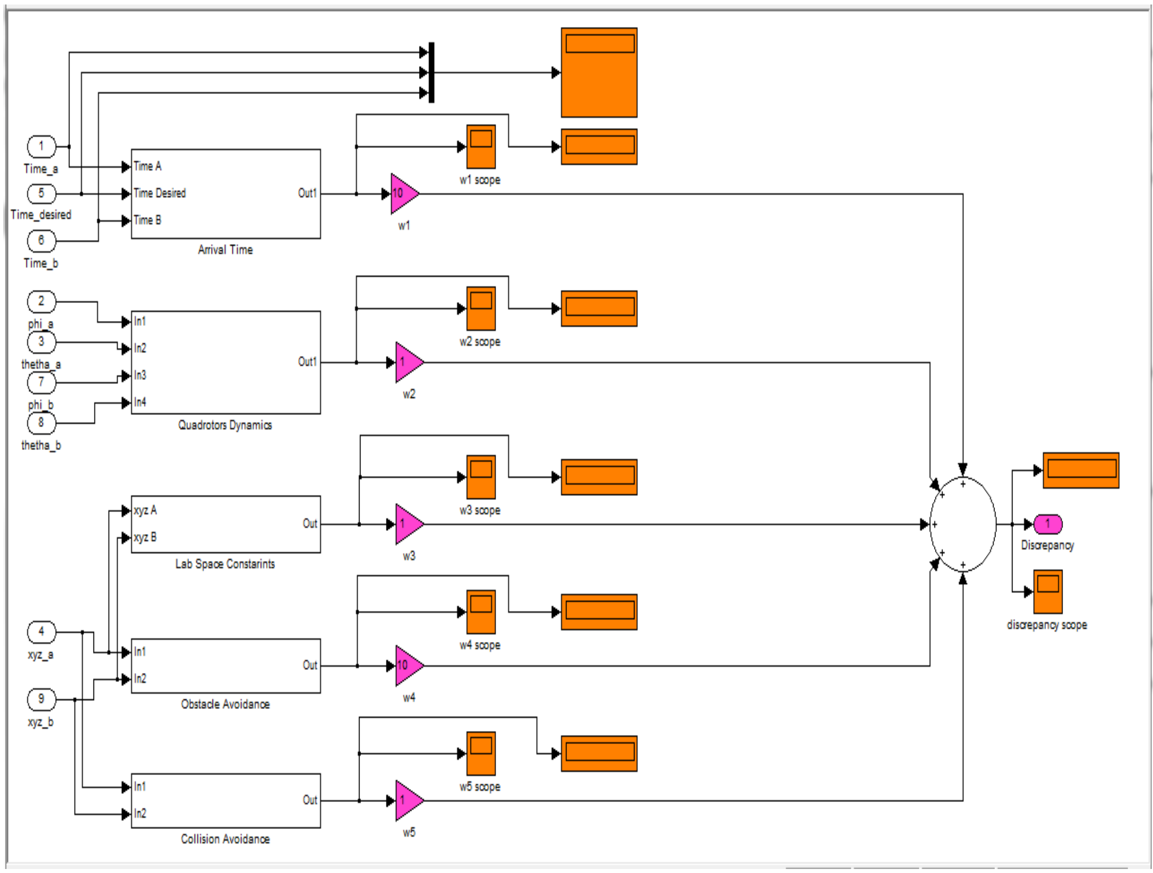


Figure 30. Cost function application in the discrepancies sub-model

2. Mission Execution

The mission trajectory is imported to the control modules for execution. Controlling the vehicles involves running the appropriate Simulink models on the ground station computer which in turn transmits all data to the vehicles via an adhoc wireless network. Of the Simulink models, the host model gathers data from the OptiTrack system as well as the USB joystick for motion control (Figure 31).

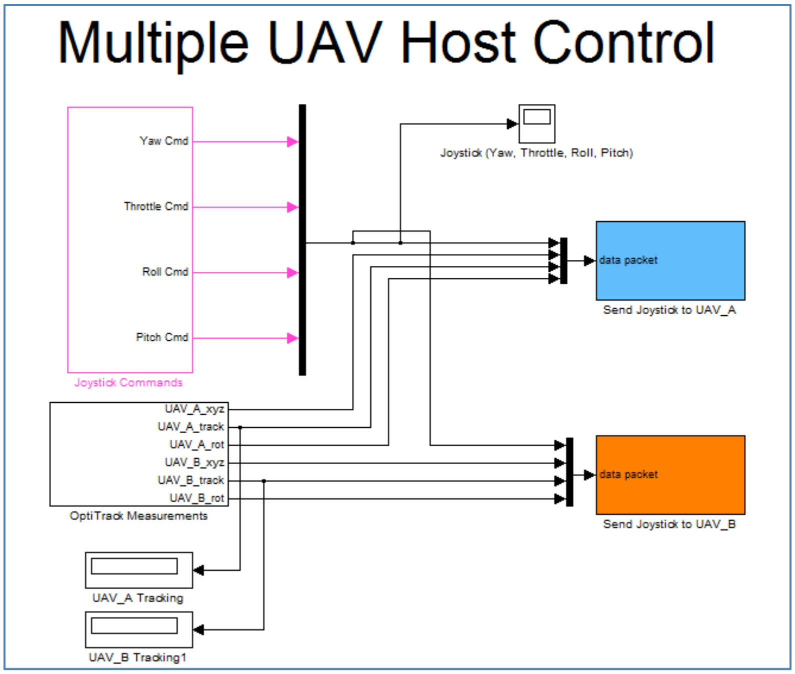


Figure 31. Host control Simulink model (Multiple UAV)

The host control module shown in Figure 31 is specifically for the multiple-UAV scenario. The module for the single UAV scene is a simpler version of what is shown. The control models (Figure 32), which are linked to each specific vehicle, compile and download code to the Gumstix processors on board the quadrotor for execution.

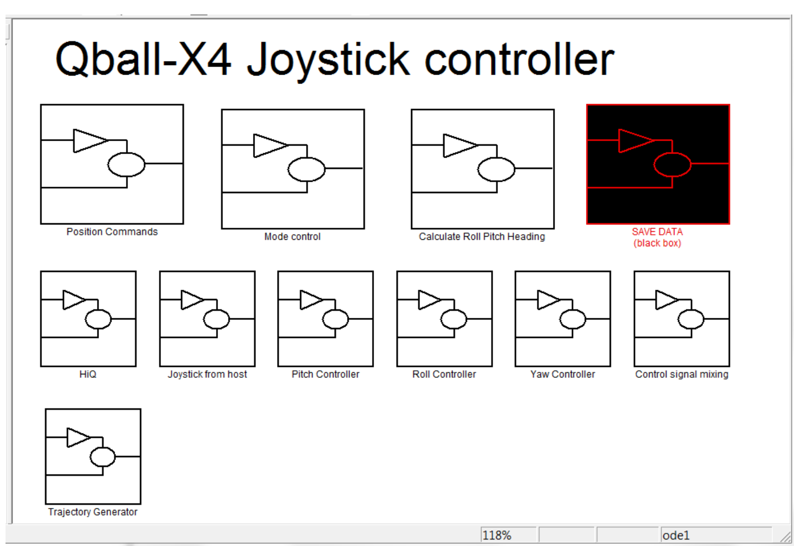


Figure 32. Quadrotor control Simulink model

Ideally, the optimization process for trajectory generation should be embedded in the quadrotor control module for seamless obstacle avoidance and mission execution. For the purpose of research however, the two processes are segregated for the ease and flexibility of tweaking and modifying the trajectory optimizing process. The procedures for performing the experiments are detailed in Appendix C.

3. Experiment Constraints

Due to the physical limitations of the indoor laboratory and the quadrotor platform, several constraints were applied for all the experiments performed. These constraints are listed as follows:

- Flight Boundary Constraints: All flights are performed within the boundary of the indoor laboratory, and is given in terms of the x, y and z dimensions. $(-2m < x < 2m), (-2m < y < 2m), (0.3m < z < 1.5m)$
- Roll and Pitch Constraints: Maximum roll and pitch angles of the quadrotor was maintained at < 0.2 radians.
- Master take-off and landing commands are given manually using the joystick control for added safety. Anytime any abnormality is observed, the experiment will be aborted.

C. SCENE 1

A single unmanned system is tasked to follow a pre-planned trajectory and scan the debris horizon for surviving casualties. While scanning, the UAV encounters an obstacle that prevents the UAV from proceeding forward. This obstacle requires the UAV to make flight adjustments to negotiate over and above the obstacle.

1. Input parameters

The vital inputs prior to the generation of the avoidance trajectory are listed in this section. The desired time taken to negotiate the obstacle, the initial and final kinematic boundary conditions are determined and entered into the Simulink model. The values are shown in Table 5.

Table 5. Input Parameters for Scene 1

Parameter	Value
Time desired, t_{des}	25 seconds
Initial Location, $x(t_0), y(t_0), z(t_0)$	1.5m, 0m, 0.5m
Initial Velocity, $\dot{x}(t_0), \dot{y}(t_0), \dot{z}(t_0)$	0 m/s,0 m/s ,0 m/s
Initial Acceleration, $\ddot{x}(t_0), \ddot{y}(t_0), \ddot{z}(t_0)$	0 m/s^2,0 m/s^2 ,0 m/s^2
Final Location, $x(t_f), y(t_f), z(t_f)$	-1.5m, 0m, 0.5m
Final Velocity, $\dot{x}(t_f), \dot{y}(t_f), \dot{z}(t_f)$	0 m/s,0 m/s ,0 m/s
Final Acceleration, $\ddot{x}(t_f), \ddot{y}(t_f), \ddot{z}(t_f)$	0 m/s^2,0 m/s^2 ,0 m/s^2

Due to the limited space of the indoor laboratory, the initial and final kinematic conditions are assumed to be zero. In actual operations, these values would be the actual kinematic values of the platform before and after entering the obstacle avoidance maneuver. Prior to generating the trajectory, the "initial guess" for the direct method virtual parameters are entered into the optimization script.

2. Results

The trajectory was generated in the MATLAB interpretative environment and the resulting varied parameters computed are presented in Table 6.

Table 6. Computed varied parameter

Varied Parameter	Value
λ_i''	0.01
λ_f''	0.0105
x_i'''	0.006
α_x	0
x_f'''	0.006
$x_f'''\alpha_x$	0
τ_f	0.035
$x_i'''\theta_x$	1.309
$x_f'''\theta_x$	-1.309

The trajectory generated by the Simulink model is pictorially presented by Figure 33. As prescribed in the initial and final conditions, the UAV starts its maneuver on the right of the room, makes its way over and above the obstacle in the middle of the room, and finally reached the left of the room. The green square in the figure are the boundary of the vertical obstacle and the yellow circle marks the “no-go zone” or avoidance boundary to be adhered by the UAV. This yellow boundary is added as a safety buffer to account for disturbances and it ensures that UAV will safely clear the obstacle.

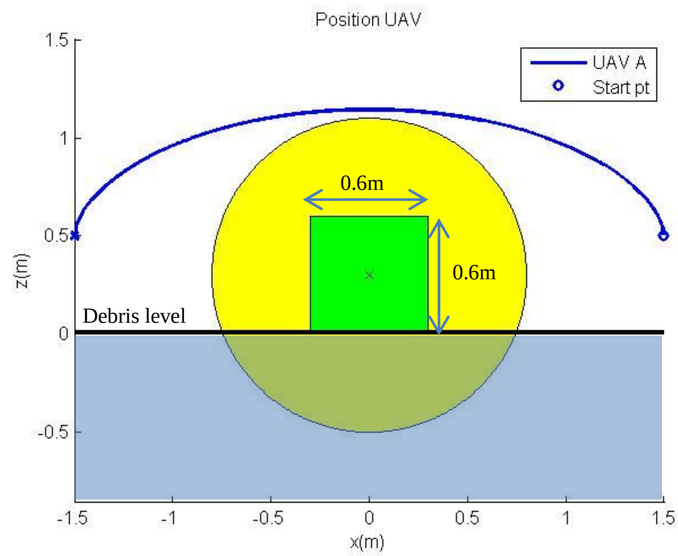


Figure 33. Trajectory generated for Scene 1

Figure 34 presents the speed factor and the speed of the UAV in Scene 1. This trajectory was performed by the UAV in the experiment and the results will follow in the next section.

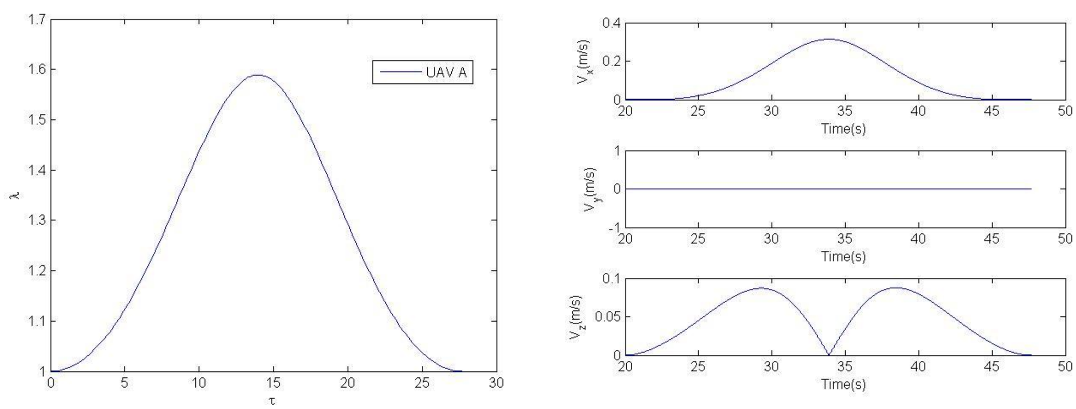


Figure 34. Speed factor (lambda) and speed for Scene 1

Using the procedures laid out in the Mission Execution section, the trajectory is performed by the quadrotor UAV in the indoor laboratory. In Figure 35, the trajectory of the actual flight is plotted against the commanded trajectory. Figure 36 details the

deviation of the commanded trajectory and the actual flight data in their respective flight axis. Finally, Figure 37 showcases the deviations in the roll and pitch (Euler angles) parameters.

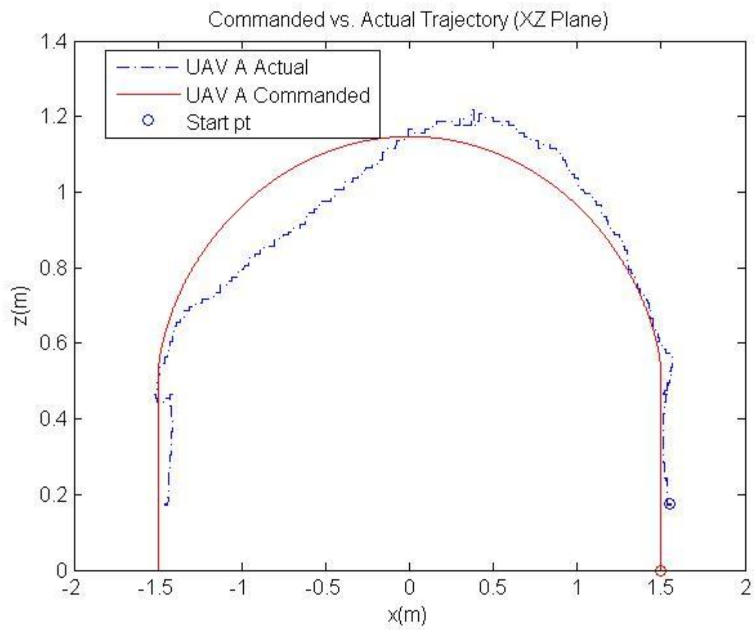


Figure 35. Commanded vs. Actual trajectory (Scene 1)

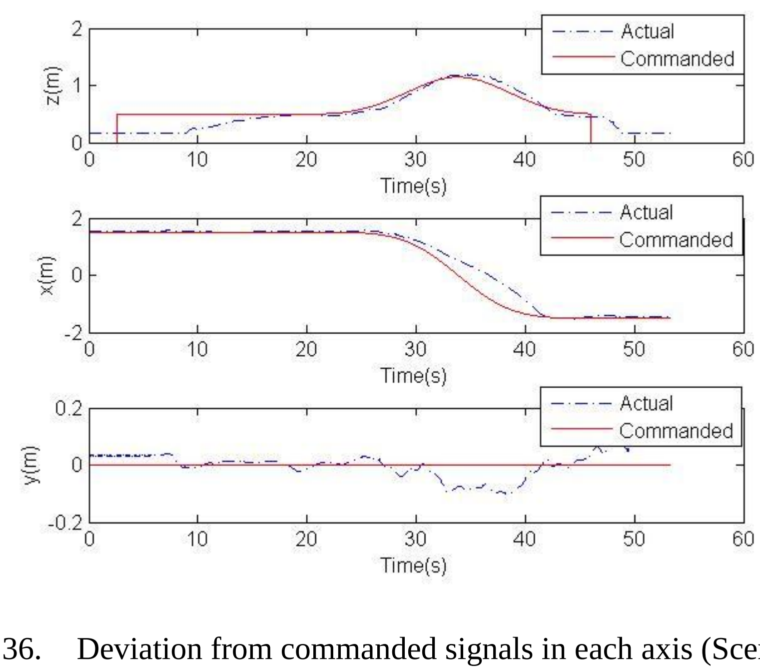


Figure 36. Deviation from commanded signals in each axis (Scene 1)

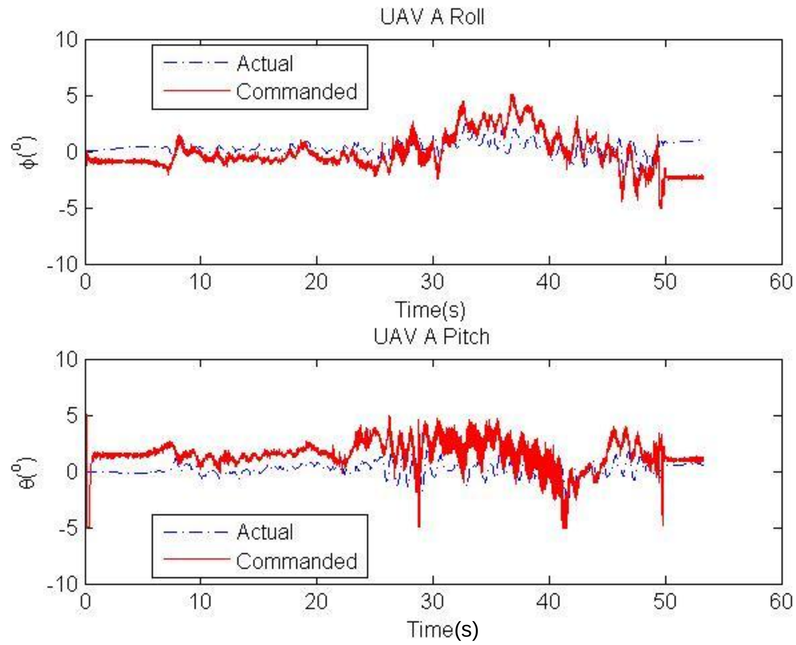


Figure 37. Euler angles of the UAV during flight (Scene 1)

It is evident from the flight data that the generated trajectory was successfully performed by the quadrotor. The discrepancy observed between the commanded trajectory and actual flight was expected. This is due to the disturbances caused by the environment on the flight performance of the quadrotor. The safety buffer that was accounted in the generation of the trajectory serves to provide added assurance that the actual flight would indeed be collision free.

D. SCENE 2

In Scene 2, multiple UAVs have been deployed in the field to perform auxiliary duties that includes searching for survivors, structural inspection, communication beacon and etc. In particular, 2 UAVs while maneuvering in the cluttered environment are headed toward each other in a collision course. Besides avoiding each other, they will have to simultaneously avoid a raised barrier that is blocking their advancements in their respective trajectory.

1. Input parameters

The vital inputs prior to the generation of the avoidance trajectory are listed in this section. The desired time taken to negotiate the obstacle, the initial and final kinematic boundary conditions are determined and entered into the Simulink model. The values are shown in Table 7.

Table 7. Input parameters for Scene 2 UAV A & UAV B

Parameter	UAV A Values	UAV B Values
Time desired, t_{des}	30 seconds	30 seconds
Initial Location, $x(t_0), y(t_0), z(t_0)$	1.5m, 0m, 0.5m	-1.5m, 0m, 0.5m
Initial Velocity, $\dot{x}(t_0), \dot{y}(t_0), \dot{z}(t_0)$	0 m/s,0 m/s ,0 m/s	0 m/s,0 m/s ,0 m/s
Initial Acceleration, $\ddot{x}(t_0), \ddot{y}(t_0), \ddot{z}(t_0)$	0 m/s^2,0 m/s^2 ,0 m/s^2	0 m/s^2,0 m/s^2 ,0 m/s^2
Final Location, $x(t_f), y(t_f), z(t_f)$	-1.5m, 0m, 0.5m	1.5m, 0m, 0.5m
Final Velocity, $\dot{x}(t_f), \dot{y}(t_f), \dot{z}(t_f)$	0 m/s,0 m/s ,0 m/s	0 m/s,0 m/s ,0 m/s
Final Acceleration, $\ddot{x}(t_f), \ddot{y}(t_f), \ddot{z}(t_f)$	0 m/s^2,0 m/s^2 ,0 m/s^2	0 m/s^2,0 m/s^2 ,0 m/s^2

Similar to Scene 1, prior to generating the trajectory, the ‘initial guess’ for the direct method virtual parameters are entered into the optimization script. The parameters are shown in Table 8.

Table 8. Varied parameters for UAV A & UAV B for Scene 2

Varied Parameter (Initial Guess)	UAV A Values	UAV B Values
λ_i''	0.01	0.01
λ_f''	0.01	0.01
x_i'''	0.01	0.01
α_x	2.182	5.323
x_f'''	0.01	0.01
$x_f'''\alpha_x$	-2.182	0.960
τ_f	0.045	0.045
$x_i'''\theta_x$	1.309	1.309
$x_f'''\theta_x$	-1.309	-1.309

2. Results

The resulting values of the varied parameters computed are presented in Table 9. The chosen parameters in the previous section resulted in the asymmetric nature of the trajectories generated for the two UAVs.

Table 9. Initial varied parameters for Scene 2

Varied Parameter	UAV A Values	UAV B Values
λ_i''	0.0095	0.0076
λ_f''	0.0116	0.0121
x_i'''	0.0125	0.0092
α_x	1.6907	5.1167
x_f'''	0.0029	0.0138
$x_f'''\alpha_x$	-2.5752	1.7593
τ_f	0.0406	0.0395
$x_i'''\theta_x$	0.6539	2.6018
$x_f'''\theta_x$	-0.9320	-0.5870

The trajectory generated by the Simulink model is pictorially presented by Figure 38. As prescribed in the initial and final conditions, the UAVs start their maneuver on either side of the room, make its way over and above the obstacle in the middle of the room, and finally reached the opposite side of where it started. The sphere in the figure represents the "no-go zone" or avoidance boundary to be adhered by the UAVs. Besides avoiding the obstacles, the UAVs have to avoid collision with each other.

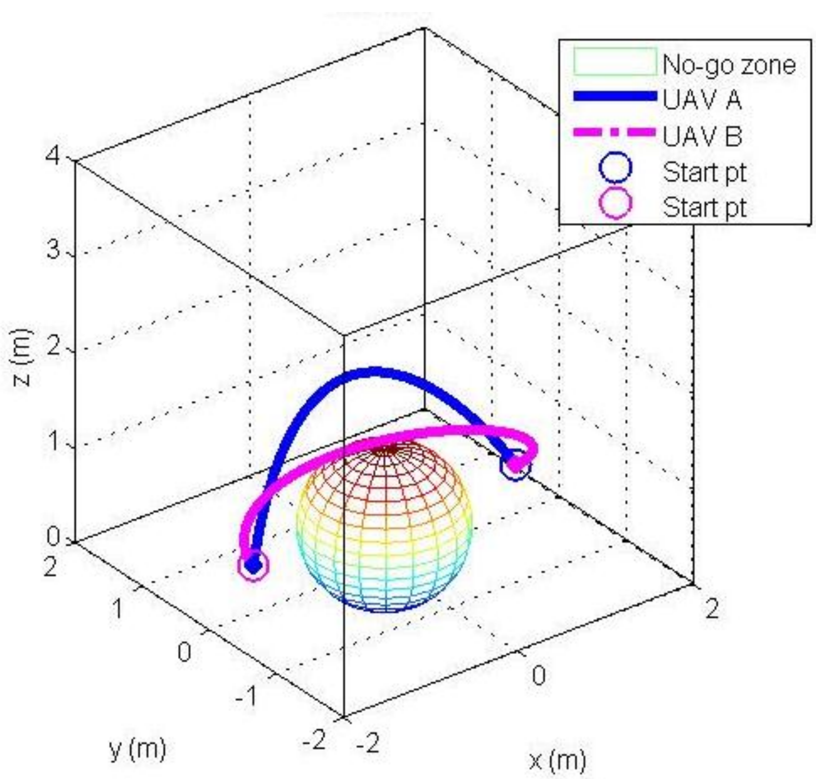


Figure 38. Trajectories generated for Scene 2

Figure 39 presents the speed factor and the speed of the two UAVs in Scene 2. The trajectories were then performed by the UAVs via actual flight to ensure its validity.

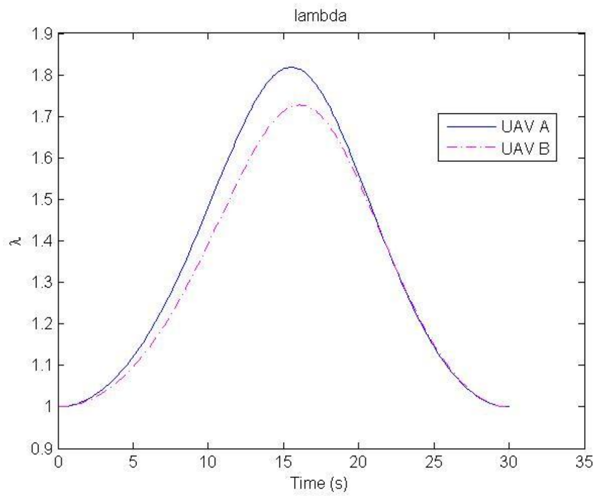


Figure 39. Speed factor (lambda) for UAV A and UAV B

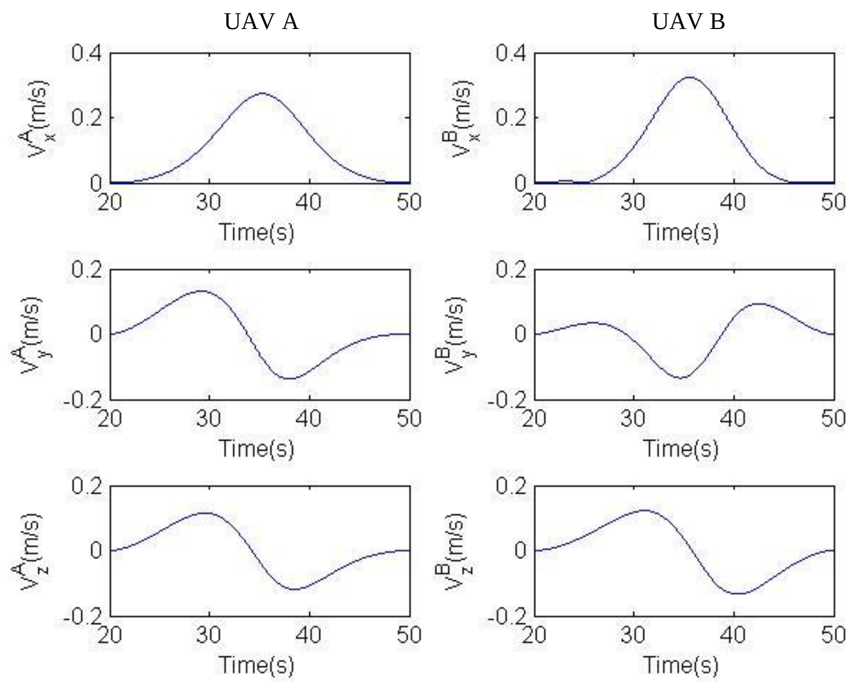


Figure 40. Speed for UAV A and UAV B

Using the procedures laid out in the Mission Execution section, the trajectory is performed by the quadrotor UAVs in the indoor laboratory. The results of the flight are presented in Figure 41, Figure 42, Figure 44, Figure 45 and Figure 46.

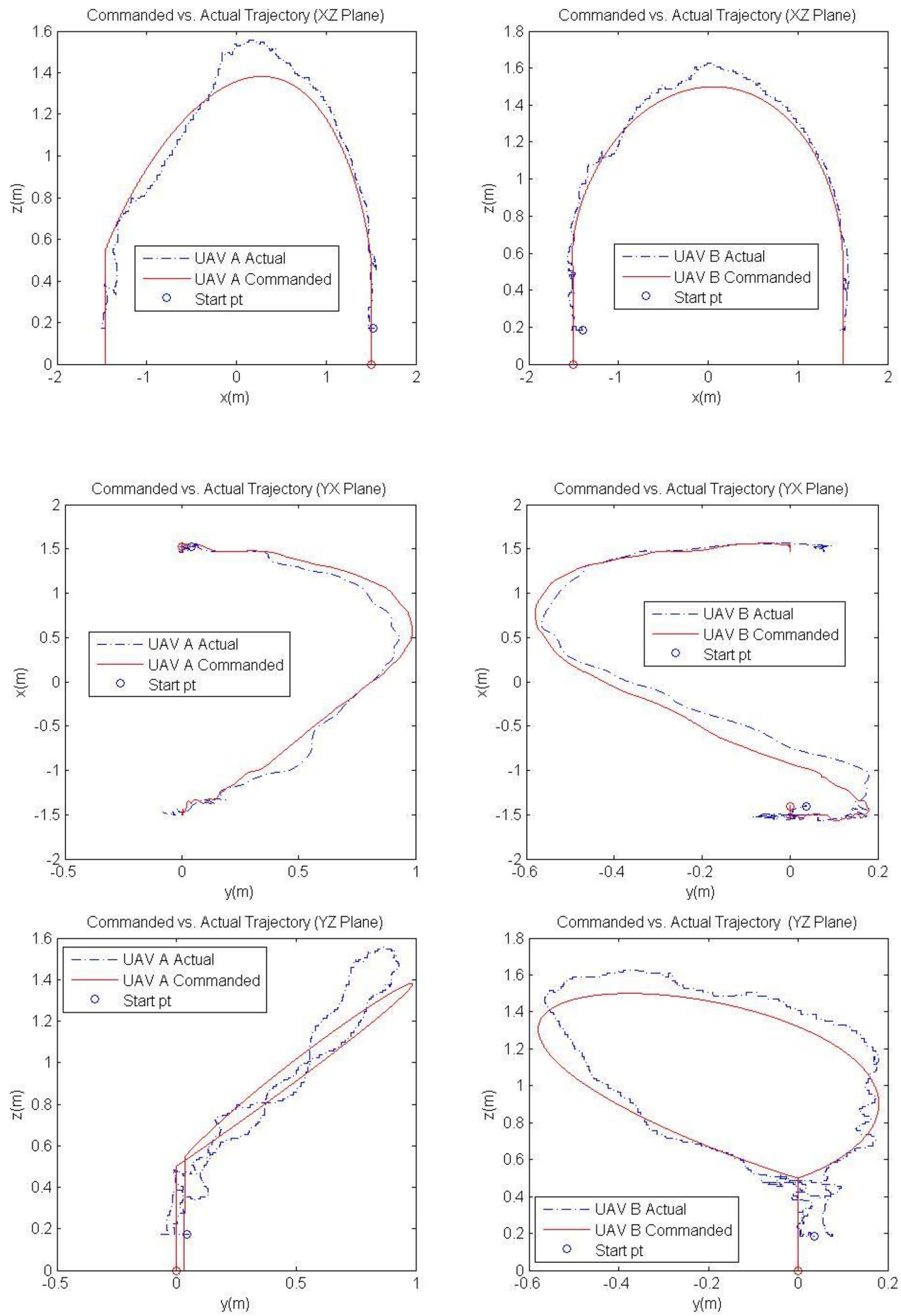


Figure 41. Commanded vs. Actual trajectories (Scene 2, various planes)

As seen from the results, the Quanser LQR controller produces about 10 cm maximum errors, which is an acceptable performance when the length of the trajectory spans in meters. In the case of this experiment, the area of operations is limited to a 2x2m space and this result in tracking errors that were significant.

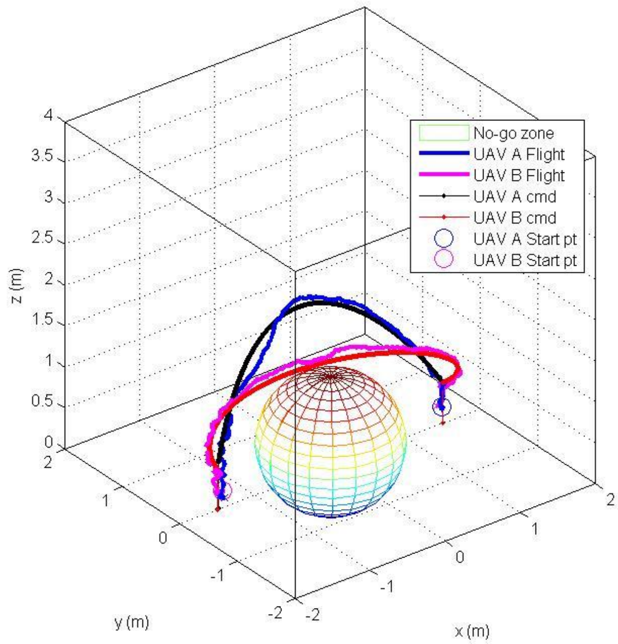


Figure 42. Commanded vs. Actual trajectory (Scene 2 – 3D)

Figures 41 and 42 present the trajectories of both UAVs. The actual flight data is plotted with the commanded trajectory to highlight the ability of the quadrotor to execute the commanded trajectories. Figure 43 represents the laboratory implementation of the trajectories involving UAV A and UAV B.

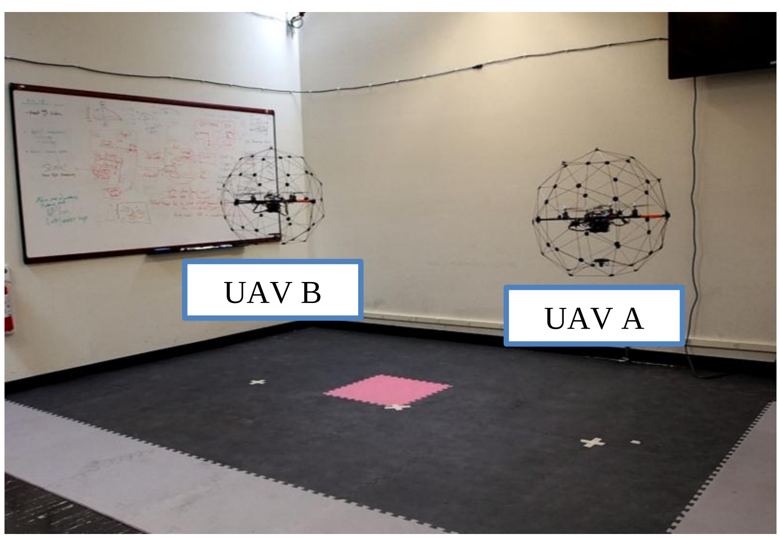


Figure 43. Laboratory implementation

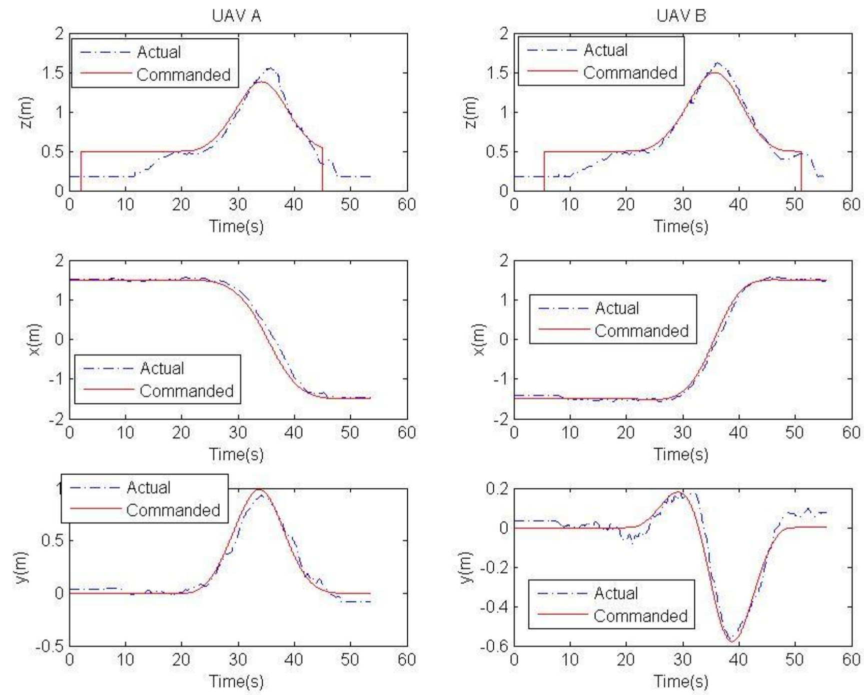


Figure 44. Deviation from commanded signals for UAV A & B in each axis (Scene 2)

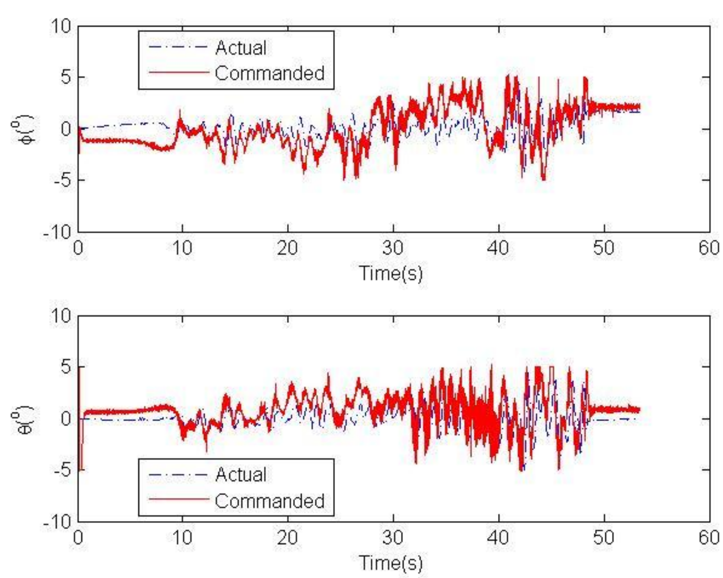


Figure 45. Euler angles of the UAV A during flight (Scene 2)

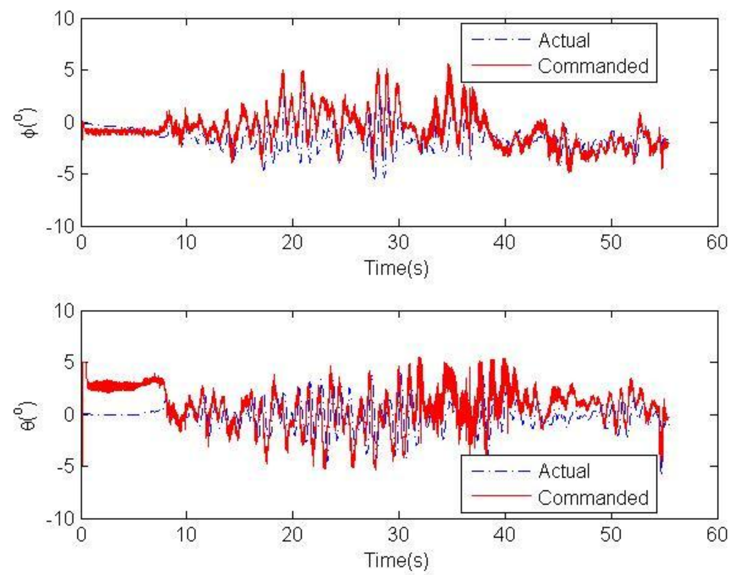


Figure 46. Euler angles of the UAV B during flight (Scene 2)

Similar to Scene 1, it is clear from the flight data that the generated trajectory was successfully and collaboratively performed by multiple quadrotors. Disturbances in the flight environment affected the flight performance of the quadrotors and resulted in the discrepancies between the commanded and actual trajectories. These discrepancies were expected and were within satisfactory margins. UAV A and UAV B were able to perform their respective trajectories without impact each other or the obstacles.

VI. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSION

The threats of disaster to the populace inhabiting urbanized areas are real. In a resource constrained environment, unmanned systems used in disaster rescue operations will free up valuable assets for other critical functions. One crucial finding from this thesis research is that in order to maximize the benefits of these unmanned systems, more autonomy during such operations is pivotal. In order for an unmanned system to be fully autonomous, a controller that incorporates both the roles of trajectory planning and trajectory following is needed. This thesis introduces such a controller to achieve the desired level of autonomy, and allow the generation of a quasi-optimal trajectory for the quadrotor platform.

Before solving the problem of generating a feasible and collision-free trajectory, systems engineering techniques were utilized to analyze the problem space and further define the effective needs and requirements of operating autonomous unmanned systems in an urban search and rescue environment. Application of unmanned vehicles was deemed to enhance the reconnaissance, intelligence and surveillance capabilities of the responding forces, while limiting the exposure risk of personnel. The limitations and constrains of unmanned systems were discussed and various important considerations were highlighted for the creation of a valid concept of operations. The envisaged thesis design scenario proof to be useful as specific operating conditions were used to develop specific sub-scenes for experimentation.

To solve the problem of trajectory generation, this thesis applied the IDVD methodology, described in Chapter IV, to achieve a desired level of autonomy. The methodology was implemented by dynamic models created using Simulink within the MATLAB interpretative environment. Models were able to rapidly generate quasi-optimal trajectories that meet the immediate needs of the platform to provide a

collision-free flight. With the use of Simulink models to generate trajectories, the designs of the desired trajectory are flexible and it offers a great learning platform for apprentices trying to learn and appreciate IDVD.

ASEIL was set up to perform the required experiments to validate and verify the feasibility of the optimization methodology. Capabilities and the limitations of the systems used to perform the experiments were identified for the investigation of two sub-scenes from the thesis design scenario. The first scene involves a single quadrotor, and the second scene required two quadrotor working collaboratively. As expected, both scenes were executed by the UAV platforms successfully. Although tracking errors were observed, it did not adversely impact the intended flight of the platforms. It can be seen that the LQR controller produces about 10 cm maximum errors, which is an acceptable performance when the length of the trajectory spans in meters. The experiments that were conducted verified the cogency of the generated trajectories and validated the unmanned systems' ability to avoid obstacles and carry out collaborative missions successfully.

B. RECOMMENDATIONS

In the interest of similar area of studies, the following recommendations for future works are:

- Tweak and improve the response of the quadrotor platform so that the reference trajectories generated can be applied and followed seamlessly.
- Creation of a simulated environment in which the optimized trajectories can be performed by a virtual system in a simulated environment before testing on an actual platform.
- Incorporation of dynamic sensors onto the UAV platform to allow observation and tracking of obstacles. Applicable sensors could be visual sensors, infrared or ultrasound sensors. In tandem, develop tracking algorithm and obstacle avoidance strategies.
- Expand the area of the experiment and incorporate more unmanned systems in operations. A larger indoor laboratory or a controlled outdoor environment could be utilized.
- Enable trajectory updates onboard the unmanned systems through the use of advanced codification of trajectory generation algorithm.

APPENDIX A

A. CAMERA LOCATION SCRIPT (MICROSOFT VISUAL C++)

```
// OptitrackCameraLocations.cpp : Defines the entry point for the
console application. Using Visual C++ 2008 Express Edition
//
#include "stdafx.h"
#include "conio.h"
#include "NPTrackingTools.h"

int _tmain(int argc, _TCHAR* argv[])
{
	int i;
	int num_cameras;
	int npresult = TT_Initialize();
	if (npresult != NPRESULT_SUCCESS) {
		printf("ERROR initializing TrackingTools: %s\n,"
TT_GetResultString(npresult));
		return 0;
	}
	npresult = TT_LoadCalibration("calibration.cal");
	if (npresult != NPRESULT_SUCCESS) {
		printf("ERROR loading TT calibration: %s\n,"
TT_GetResultString(npresult));
		TT_FinalCleanup();
		TT_Shutdown();
		return 0;
	}
	num_cameras = TT_CameraCount();
	printf("There are %d cameras connected and calibrated.\n,"
num_cameras);

	TT_Update();

	for (i = 0; i < num_cameras; i++) {
		// Get camera i location X, Y, Z
		float x, y, z;
		x = TT_CameraXLocation(i);
		y = TT_CameraYLocation(i);
		z = TT_CameraZLocation(i);

		printf("Camera %d [%s] location (X, Y, Z) =
(%f, %f, %f)\n," i+1, TT_CameraName(i), x, y, z);
	}
	printf("Done. Press ENTER to quit.\n");
	getch();

	TT_FinalCleanup();
	TT_Shutdown();

	return 0;
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B

A. OPTIMIZATION SCRIPT FOR DIRECT METHOD (SINGLE UAV)

```
close all, clear all, clc
warning off
D2R=3.14159/180;

%% Mission Inputs
global vert_gain time_des
global a0XYZ a0XYZd a0XYZ2d afXYZ afXYZd afXYZ2d
vert_gain = 1;              % vert_gain=0 corresponds to a vertical
maneuver
time_des = 30;              % Tdes desired time of mission
a0XYZ   = [ 1.5; 0; 0.5];  % initial position for Quadrotor a
a0XYZd  = [   0; 0;   0];  % initial velocity for Quadrotor a
a0XYZ2d = [   0; 0;   0];  % initial acceleration for Quadrotor a
afXYZ   = [-1.5; 0; 0.5];  % final position for Quadrotor a
afXYZd  = [   0; 0;   0];  % final velocity for Quadrotor a
afXYZ2d = [   0; 0;   0];  % final acceleration for Quadrotor a

%% Initial Guess for Varied Parameters
x0=[0.01            % lam0_2pr_a
    0.01            % lamf_2pr_a
    0.01            % X0a_tpl_prime
    125*D2R         % X0a_tpl_prime_angle, radians
    0.01            % Xfa_tpl_prime
    -125*D2R        % Xfa_tpl_prime_angle, radians
    45/1000         % tauf_a
    75*D2R          % X0a_tpl_prime_vert_angle, radians
    -75*D2R         % Xfa_tpl_prime_vert_angle, radians

t = cputime;
options=optimset('TolFun', 1e-1, 'TolX', 1e-1, 'Display', 'final');
[x0,fval,exitflag,output]=fminsearch(@DMlfun,x0,options)
time_elapsed = cputime-t

lam0_2pr_a=x0(1);
lamf_2pr_a=x0(2);
X0a_tpl_prime=x0(3);
X0a_tpl_primeA=x0(4);
Xfa_tpl_prime=x0(5);
Xfa_tpl_primeA=x0(6);
tauf_a=x0(7);
X0a_tpl_primeB=x0(8);
Xfa_tpl_primeB=x0(9);

%% Compute Controls Pos

% Controller speed
ctrl_t_step = .005;
```

```
% Run Simulation to get data
sim('DM3', [0 200])
[m_a,n_a] = size(a);
t_a_end = a(m_a,1);
t_a = 0:ctrl_t_step:t_a_end;

% Setup Variables
tau_a = a(:,1);
phi_a = a(:,2);
theta_a = a(:,3);
x_a = a(:,4);
y_a = a(:,5);
z_a = a(:,6);
lambda_a = a(:,7);
x_vel_a = a(:,8);
y_vel_a = a(:,9);
z_vel_a = a(:,10);
x_accel_a = a(:,11);
y_accel_a = a(:,12);
z_accel_a = a(:,13);


%% Interpolate data
% Interpolate data between points at the same frequency the controller
% runs at.
phi_a = interp1(tau_a,phi_a,t_a,'pchip');
theta_a = interp1(tau_a,theta_a,t_a,'pchip');
x_a = interp1(tau_a,x_a,t_a,'pchip');
y_a = interp1(tau_a,y_a,t_a,'pchip');
z_a = interp1(tau_a,z_a,t_a,'pchip');
x_vel_a = interp1(tau_a,x_vel_a,t_a,'pchip');
y_vel_a = interp1(tau_a,y_vel_a,t_a,'pchip');
z_vel_a = interp1(tau_a,z_vel_a,t_a,'pchip');
x_accel_a = interp1(tau_a,x_accel_a,t_a,'pchip');
y_accel_a = interp1(tau_a,y_accel_a,t_a,'pchip');
z_accel_a = interp1(tau_a,z_accel_a,t_a,'pchip');

%% Plots
close all;

obs_x = 0; obs_z = 0; r = 0.8;  %Radius of obstacle is 0.2, radius of
quad= 0.5, safe dist = 0.1, total = 0.8m
obs_w = 0.6; obs_h = 0.6;       %Obstacle width and height
d = r*2; px = obs_x-r; pz = obs_z+0.3-r;

figure
rectangle('Position',[px pz d d],'Curvature',[1,1],'FaceColor','y');
hold on;
rectangle('Position',[(obs_x-obs_w/2) obs_z obs_w
obs_h],'Curvature',[0,0],'FaceColor','g');
hold on;
rectangle('Position',[(obs_x-1.5) (obs_z) 3
0.02],'Curvature',[0,0],'FaceColor','black');
hold on;
```

```
plot(x_a,z_a,'b-','LineWidth',2)
hold on;
plot(x_a(1,1),z_a(1,1),'bo','LineWidth',2)
hold on;
plot(x_a(end,end),z_a(end,end),'bx','LineWidth',2)
title('Position UAV')
hold on;
legend('UAV A','Start pt',1)
plot(obs_x,obs_z+obs_h/2,'blackx')
hold on;
axis equal
xlabel('x(m)')
ylabel('z(m)')

figure
plot(y_a,x_a,'b-','LineWidth',2)
hold on;
plot(y_a(1,1),x_a(1,1),'bo','LineWidth',2)
hold on;
plot(y_a(end,end),x_a(end,end),'bx','LineWidth',2)
title('Position UAV')
hold on;
legend('UAV A','Start pt',1)
set(gca, 'Ydir', 'reverse')
xlabel('y(m)')
ylabel('x(m)')

figure
plot(tau_a,lambda_a,'b')
hold on;
% plot(tau_b,lambda_b,'-.r')
xlabel('tau')
ylabel(texlabel('lambda'))
legend('UAV A',0)
title('lambda')

figure
subplot(3,1,1);
plot(t_a, abs(x_vel_a));
title('X Velocity');
xlabel('Time');ylabel('Speed');
subplot(3,1,2);
plot(t_a, abs(y_vel_a));
title('Y Velocity');
xlabel('Time');ylabel('Speed');
subplot(3,1,3);
plot(t_a, abs(z_vel_a));
title('Z Velocity');
xlabel('Time');ylabel('Speed');

%% Setup data for use in controller
% Setup a series of commands for the first waypoint
t_start = 20; %Start time for maneuver
t_a = t_a+t_start;
t_beginning = 0:ctrl_t_step:t_start-ctrl_t_step;
```

```
z_comp = ones(1,length(t_beginning));

t_comp_a = [t_beginning' t_beginning';t_a' t_a'];
x_command_a = [t_beginning' x_a(1)*z_comp';t_a' x_a'];
y_command_a = [t_beginning' y_a(1)*z_comp';t_a' y_a'];
z_command_a = [t_beginning' z_a(1)*z_comp';t_a' z_a'];
```

B. OPTIMIZATION SCRIPT FOR DIRECT METHOD (MULTIPLE UAV)

```
close all, clear all, clc
warning off
D2R=3.14159/180;

%% Mission Inputs
global vert_gain time_des
global a0XYZ a0XYZd a0XYZ2d afXYZ afXYZd afXYZ2d
global b0XYZ b0XYZd b0XYZ2d bfXYZ bfXYZd bfXYZ2d
vert_gain = 1;            % vert_gain=0 corresponds to a vertical
maneuver
time_des = 30;            % Tdes desired time of mission
a0XYZ   = [ 1.5; 0; 0.5];  % initial position for Quadrotor a
a0XYZd  = [   0; 0;   0];  % initial velocity for Quadrotor a
a0XYZ2d = [   0; 0;   0];  % initial acceleration for Quadrotor a
afXYZ   = [-1.5; 0; 0.5];  % final position for Quadrotor a
afXYZd  = [   0; 0;   0];  % final velocity for Quadrotor a
afXYZ2d = [   0; 0;   0];  % final acceleration for Quadrotor a
b0XYZ   = [-1.5; 0; 0.5];  % initial position for Quadrotor b
b0XYZd  = [   0; 0;   0];  % initial velocity for Quadrotor b
b0XYZ2d = [   0; 0;   0];  % initial acceleration for Quadrotor b
bfXYZ   = [ 1.5; 0; 0.5];  % final position for Quadrotor b
bfXYZd  = [   0; 0;   0];  % final velocity for Quadrotor b
bfXYZ2d = [   0; 0;   0];  % final acceleration for Quadrotor b

%% Initial Guess for Varied Parameters
x0=[0.01             % lam0_2pr_a
    0.01             % lamf_2pr_a
    0.01             % X0a_tpl_prime
    125*D2R          % X0a_tpl_prime_angle, radians
    0.01             % Xfa_tpl_prime
    -125*D2R         % Xfa_tpl_prime_angle, radians
    45/1000          % tauf_a
    75*D2R           % X0a_tpl_prime_vert_angle, radians
    -75*D2R          % Xfa_tpl_prime_vert_angle, radians
    0.01             % lam0_2pr_b
    0.01             % lamf_2pr_b
    0.01             % X0b_tpl_prime
    (125+180)*D2R    % X0b_tpl_prime_angle, radians
    0.01             % Xfb_tpl_prime
    (-125+180)*D2R   % Xfb_tpl_prime_angle, radians
    45/1000          % tauf_b
    75*D2R           % X0b_tpl_prime_vert_angle, radians
    -75*D2R];        % Xfb_tpl_prime_vert_angle, radians

%% Optimization
```

```
t = cputime;
options=optimset('TolFun',1e-1,'TolX',1e-
1,'Display','iter')%,'MaxIter',100);
%options=optimset('TolFun',1e-1,'TolX',1e-1,'Display','final');
[x0,fval,exitflag,output]=fminsearch(@DMlfun,x0,options)
%[x0,fval,exitflag,output]=fminunc(@DMlfun,x0,options)
time_elapsed = cputime-t

lam0_2pr_a=x0(1);
lamf_2pr_a=x0(2);
X0a_tpl_prime=x0(3);
X0a_tpl_primeA=x0(4);
Xfa_tpl_prime=x0(5);
Xfa_tpl_primeA=x0(6);
tauf_a=x0(7);
X0a_tpl_primeB=x0(8);
Xfa_tpl_primeB=x0(9);
lam0_2pr_b=x0(10);
lamf_2pr_b=x0(11);
X0b_tpl_prime=x0(12);
X0b_tpl_primeA=x0(13);
Xfb_tpl_prime=x0(14);
Xfb_tpl_primeA=x0(15);
tauf_b=x0(16);
X0b_tpl_primeB=x0(17);
Xfb_tpl_primeB=x0(18);

%% Compute Controls Pos
% Controller speed
ctrl_t_step = .005;

% Run Simulation to get data
sim('DM3', [0 200])
[m_a,n_a] = size(a);
t_a_end = a(m_a,1);
t_a = 0:ctrl_t_step:t_a_end;

[m_b,n_b] = size(b);
t_b_end = b(m_b,1);
t_b = 0:ctrl_t_step:t_b_end;

% Setup Variables
time_a = a(:,1);
phi_a = a(:,2);
theta_a = a(:,3);
x_a = a(:,4);
y_a = a(:,5);
z_a = a(:,6);
lambda_a = a(:,7);
x_vel_a = a(:,8);
y_vel_a = a(:,9);
z_vel_a = a(:,10);
x_accel_a = a(:,11);
y_accel_a = a(:,12);
```

```
z_accel_a = a(:,13);

time_b = b(:,1);
phi_b = b(:,2);
theta_b = b(:,3);
x_b = b(:,4);
y_b = b(:,5);
z_b = b(:,6);
lambda_b = b(:,7);
x_vel_b = b(:,8);
y_vel_b = b(:,9);
z_vel_b = b(:,10);
x_accel_b = b(:,11);
y_accel_b = b(:,12);
z_accel_b = b(:,13);

%% Interpolate data
% Interpolate data between points at the same frequency the controller
% runs at.
phi_a = interp1(time_a,phi_a,t_a,'pchip');
theta_a = interp1(time_a,theta_a,t_a,'pchip');
x_a = interp1(time_a,x_a,t_a,'pchip');
y_a = interp1(time_a,y_a,t_a,'pchip');
z_a = interp1(time_a,z_a,t_a,'pchip');
x_vel_a = interp1(time_a,x_vel_a,t_a,'pchip');
y_vel_a = interp1(time_a,y_vel_a,t_a,'pchip');
z_vel_a = interp1(time_a,z_vel_a,t_a,'pchip');
x_accel_a = interp1(time_a,x_accel_a,t_a,'pchip');
y_accel_a = interp1(time_a,y_accel_a,t_a,'pchip');
z_accel_a = interp1(time_a,z_accel_a,t_a,'pchip');

phi_b = interp1(time_b,phi_b,t_b,'pchip');
theta_b = interp1(time_b,theta_b,t_b,'pchip');
x_b = interp1(time_b,x_b,t_b,'pchip');
y_b = interp1(time_b,y_b,t_b,'pchip');
z_b = interp1(time_b,z_b,t_b,'pchip');
x_vel_b = interp1(time_b,x_vel_b,t_b,'pchip');
y_vel_b = interp1(time_b,y_vel_b,t_b,'pchip');
z_vel_b = interp1(time_b,z_vel_b,t_b,'pchip');
x_accel_b = interp1(time_b,x_accel_b,t_b,'pchip');
y_accel_b = interp1(time_b,y_accel_b,t_b,'pchip');
z_accel_b = interp1(time_b,z_accel_b,t_b,'pchip');

%% Plots
close all;

obs_x = 0; obs_z = 0; r = 0.8;  %Radius of obstacle is 0.4, radius of
quad= 0.5, safe dist = 0.1, total = 1m
obs_w = 0.6; obs_h = 0.6;       %Obstacle width and height
d = r*2; px = obs_x-r; pz = obs_z+0.3-r;

figure
rectangle('Position',[px pz d d],'Curvature',[1,1],'FaceColor','y');
hold on;
```

```
rectangle('Position',[(obs_x-obs_w/2) obs_z obs_w
obs_h],'Curvature',[0,0],'FaceColor','g');
hold on;
rectangle('Position',[(obs_x-1.5) (obs_z) 3
0.02],'Curvature',[0,0],'FaceColor','black');
hold on;
% set(gca, 'Xdir', 'reverse')
plot(x_a,z_a,'b-','LineWidth',2)
hold on;
plot(x_b,z_b,'m-.','LineWidth',2)
hold on;
plot(x_a(1,1),z_a(1,1),'bo','MarkerSize',12)
hold on;
plot(x_b(1,1),z_b(1,1),'mo','MarkerSize',12)
hold on;
plot(x_a(end,end),z_a(end,end),'bx','LineWidth',2)
hold on;
plot(x_b(end,end),z_b(end,end),'mx','LineWidth',2)
hold on;
title('Position of 2 UAV')
hold on;
legend('UAV A','UAV B','Start pt','Start pt',1)
plot(obs_x,obs_z+obs_h/2,'blackx')
hold on;
axis equal
xlabel('x (m)')
ylabel('z (m)')

figure
plot(y_a,x_a,'b-','LineWidth',2)
hold on;
plot(y_b,x_b,'m-.','LineWidth',2)
hold on;
plot(y_a(1,1),x_a(1,1),'bo','LineWidth',2)
hold on;
plot(y_b(1,1),x_b(1,1),'mo','LineWidth',2)
hold on;
plot(y_a(end,end),x_a(end,end),'bx','LineWidth',2)
hold on;
plot(y_a(end,end),x_a(end,end),'bx','LineWidth',2)
hold on;
title('Position UAV')
hold on;
legend('UAV A','UAV B','Start pt','Start pt',1)
axis equal
xlabel('y (m)')
ylabel('x (m)')

figure
plot(time_a,lambda_a,'b')
hold on;
plot(time_b,lambda_b,'-.m')
plot(time_des*[1 1],[1 1.2],'r')
legend('UAV A','UAV B','RQM',0)
xlabel('Time (s)'), ylabel('\lambda')
```

```
title('lambda')

figure
subplot(3,2,1);
plot(t_a, abs(x_vel_a));
title('UAV A X Velocity');
xlabel('Time');ylabel('Speed');
subplot(3,2,3);
plot(t_a, abs(y_vel_a));
title('UAV A Y Velocity');
xlabel('Time');ylabel('Speed');
subplot(3,2,5);
plot(t_a, abs(z_vel_a));
title('UAV A Z Velocity');
xlabel('Time');ylabel('Speed');
subplot(3,2,2);
plot(t_b, abs(x_vel_b));
title('UAV B X Velocity');
xlabel('Time');ylabel('Speed');
subplot(3,2,4);
plot(t_b, abs(y_vel_b));
title('UAV B Y Velocity');
xlabel('Time');ylabel('Speed');
subplot(3,2,6);
plot(t_b, abs(z_vel_b));
title('UAV B Z Velocity');
xlabel('Time');ylabel('Speed');

figure;
[x,y,z] = sphere;
mesh(r*x,r*y,r*z+0.4)
hold on;
plot3(x_a, y_a, z_a,'b','linewidth',3);
box on;
plot3(x_b, y_b, z_b,'m-.','linewidth',3);
box on;
plot3(x_a(1,1),y_a(1,1),z_a(1,1),'bo','MarkerSize',12)
hold on;
plot3(x_b(1,1),y_b(1,1),z_b(1,1),'mo','MarkerSize',12)
hold on;
plot3(x_a(end,end),y_a(end,end),z_a(end,end),'bx','LineWidth',5)
grid on;
plot3(x_b(end,end),y_b(end,end),z_b(end,end),'mx','LineWidth',5)
grid on;
legend('No-go zone','UAV A','UAV B','Start pt','Start pt',1)
% Set the viewing angle and the axis limits
%view(30, 42);
axis([-2 2 -2 2 0 4]);
axis square;
view(3);
% Add title and axis labels
xlabel('x (m)');
ylabel('y (m)');
zlabel('z (m)');
title('3D view');
```

```
%% Setup data for use in controller
% Setup a series of commands for the first waypoint
t_start = 20; %Start time for maneuver
t_a = t_a+t_start;
t_b = t_b+t_start;
t_beginning = 0:ctrl_t_step:t_start-ctrl_t_step;
z_comp = ones(1,length(t_beginning));

t_comp_a = [t_beginning' t_beginning';t_a' t_a'];
x_command_a = [t_beginning' x_a(1)*z_comp';t_a' x_a'];
y_command_a = [t_beginning' y_a(1)*z_comp';t_a' y_a'];
z_command_a = [t_beginning' z_a(1)*z_comp';t_a' z_a'];

t_comp_b = [t_beginning' t_beginning';t_b' t_b'];
x_command_b = [t_beginning' x_b(1)*z_comp';t_b' x_b'];
y_command_b = [t_beginning' y_b(1)*z_comp';t_b' y_b'];
z_command_b = [t_beginning' z_b(1)*z_comp';t_b' z_b'];
```

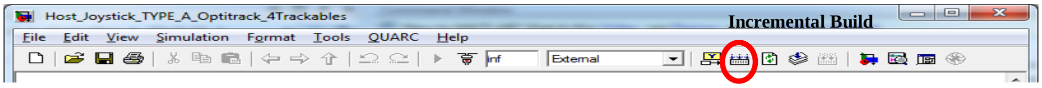
THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C

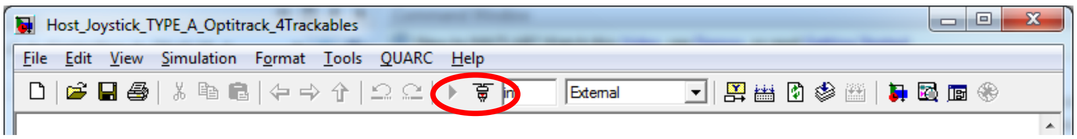
A. EXECUTING A SINGLE UAV MANEUVER

1. Ensure optitrack cameras have been calibrated, and the trackables involved in this experiment assigned. (Refer to Quanser's documentation on calibrating cameras for more information)
2. Make sure that the batteries are fully charged. (Batteries run out fairly quickly, usually after 5–6 single UAV routines. Performance is adversely affected by low batteries.)
3. Fix 2 batteries on QBall.
4. Place quadrotor on the starting point with the back of the quad (side with orange label) pointing toward the workstation.

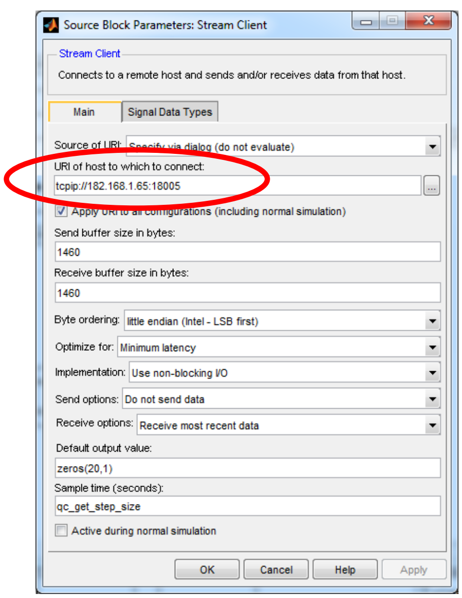
5. Check that the wireless dongle and joystick controller are connected to the PC.
6. Open the models required to perform the flight:
   i) Host_Joystick_TYPE_A_Optitrack_4_V4.mdl (Joystick and optitrack model)
   ii) Qball_x4_V4.mdl (Quad control model)
7. Go to Model (i), double-click the block "OptiTrack Measurements," then double-click block "OptiTrack Trackables."
8. A dialog box as shown below appears. Under Calibration File > Select the ."cal" calibration file generated from the Optitrack camera calibration performed. Repeat for Trackables definition file for ."tra" file.
9. Ensure that the port number in the "Send Joystick to Qball.X4" block is the same as the one in model ii), in this case, it is "tcpip://localhost:18005."
10. Go back to Model (i), click on "Incremental Build" to build the C-codes for the optitrack model.



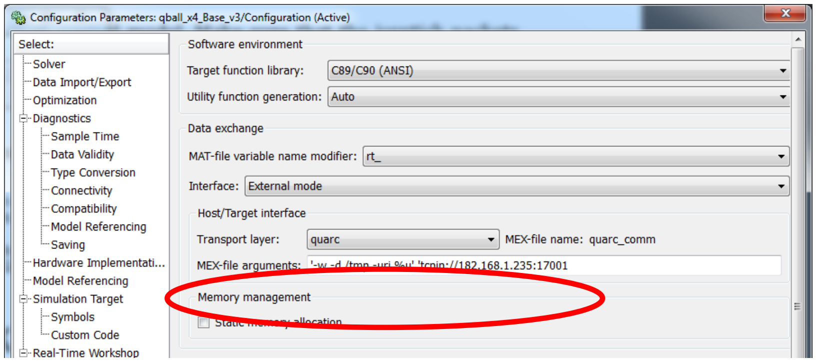
11. Once the model is built, connect and run the model. Click on "Connect to Target" and followed by "Start real-time code."



12. Check that the joystick is connected by checking on the scope for Joystick.
13. Check QBall is connected by checking that the trackable block is showing "1."
14. Connect the batteries and switch on the Qball. There will be consistent "Beep-Beep" sounds which will persist until the codes are written on the Qball.
15. Click on wireless icon on the desktop pane and connect to the wireless network "GSAH." (Refresh and wait a moment for the network to appear)
16. Back on the desktop and in Model ii), Double-click the block "Joystick from host" and followed by "Stream Client." A dialog box as shown below should appear.



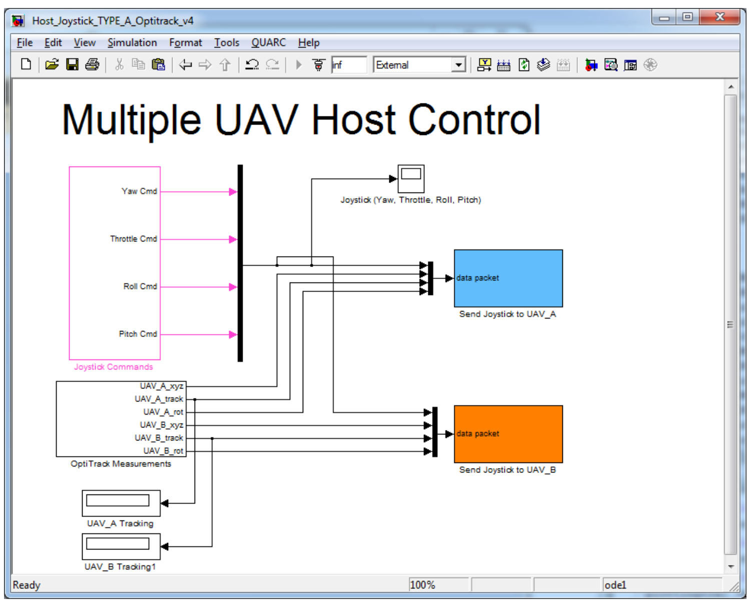
17. Check that the URI tcpip address should synchronize to the computer IP address. This can be verified by Start > "Under Search: type cmd" then press "Enter." Type ipconfig and check the IPv4 Address. Enter the specific URI into the model in the URI entry as shown in the picture above.
18. Go to Model (ii) and go to QUARC > Options on the menu list. The dialog as shown below appear



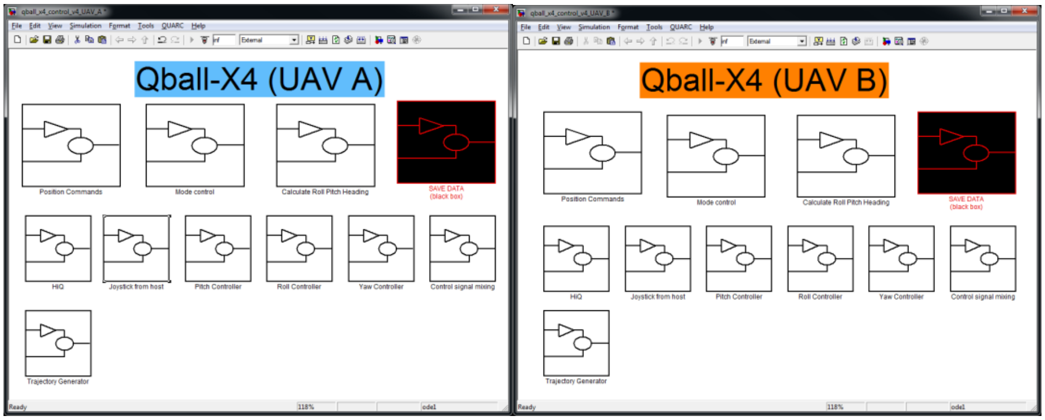
19. Check that the arguments ip address matches the QBall. There is a sticker on QBall rod to indicate its IP address.
20. Load the scene waypoints for the quad (e.g. “Scene 1.mat”). This contains the flight trajectory information.
21. Open the “console for all” panel to monitor the build process. Go to Quarc>console for all. A DOS screen will pop up if the qball is communicating with the host machine. If not, go back to step 15, and make sure that the connection is established before proceeding.
22. Click on “incremental build.” Wait for the compilation of the codes and transferring it to QBall.
23. Once built, click on “Connect to Target” and followed by “Start real-time code” to run Model (ii). The “Beep-Beep” sound should stop.
24. Start the QBall by moving the joystick’s throttle stick up to start the flight. For trajectory following, remember to start the flight soon after clicking on “start real time code.” Otherwise the quad may miss the front section of the flight and only perform the remaining flight.
25. UAV perform the flight plan.
26. When the flight id completed, move the joystick’s throttle stick down to land the UAV.
27. At any point if there is a need to stop the experiment, push the throttle down to land the UAV. Avoid pressing the stop button (in the matlab model) in the middle of a flight as it will cut command to the motor immediately.
28. Switch off the system and unplug the batteries.

B. EXECUTING MULTIPLE UAVS MANEUVER

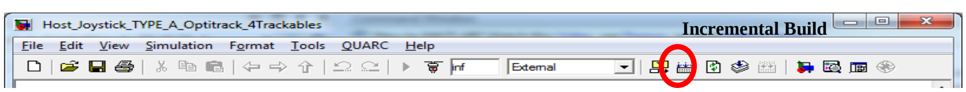
1. Ensure optitrack cameras have been calibrated, and the trackables involved in this experiment assigned. (Refer to Quanser's documentation on calibrating cameras for more information)
2. Make sure that the batteries are fully charged. (Batteries run out fairly quickly, usually after 5–6 UAV routines. Performance is adversely affected by low batteries.)
3. Fix 2 batteries on each QBall.
4. Place quadrotors at the starting positions with the back of the quad (side with orange label) pointing toward the workstation.
5. Check that the wireless dongle and joystick controller are connected to the PC.
6. Open the models required to perform the flight:
   i) Host_Joystick_TYPE_A_Optitrack_4_Multiple UAV.mdl (Joystick and optitrack model)



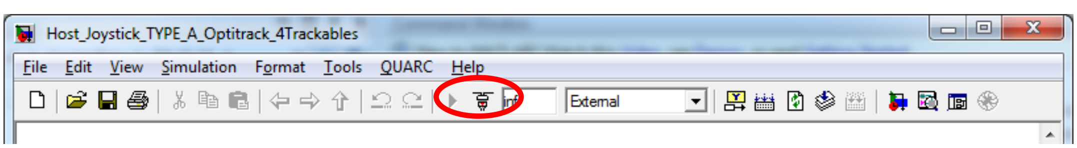
   ii) Qball_x4_V4_UAV_A.mdl (Quad control model for UAV A)
   iii) Qball_x4_V4_UAV_B.mdl (Quad control model for UAV B)



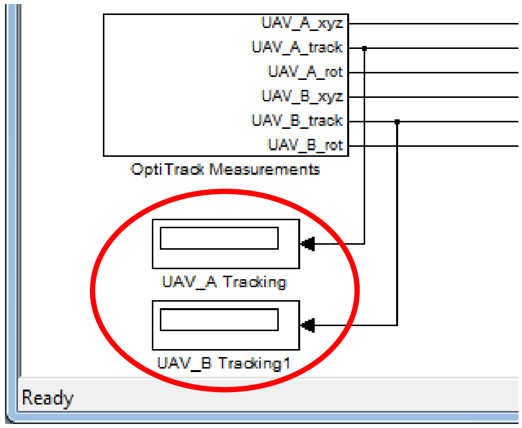
7. Go to Model (i), double-click the block “OptiTrack Measurements,” then double-click block “OptiTrack Trackables.”
8. A dialog box as shown below appears. Under Calibration File > Select the .”cal” calibration file generated from the Optitrack camera calibration performed. Repeat for Trackables definition file for .”tra” file. Define the number of trackables.
9. Back in the main Model(i), ensure that the port number in the “Send Joystick to Qball.X4” block is the unique for Model (ii) and Model(iii), in this case, it is “tcpip://localhost:18005” for UAV A and “tcpip://localhost:18006” for UAV B.
10. Go back to Model (i), click on “Incremental Build” to build the C-codes for the optitrack model.



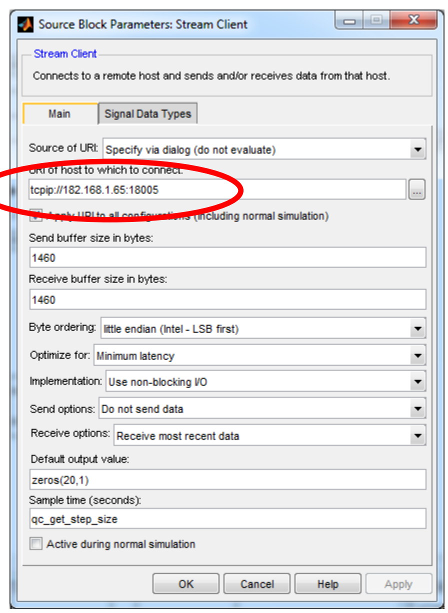
11. Once the model is built, connect and run the model. Click on “Connect to Target” and followed by “Start real-time code.”



12. Check that the joystick is connected by checking on the scope for Joystick.
13. Check QBalls are connected by checking that the trackable block is showing “1.”

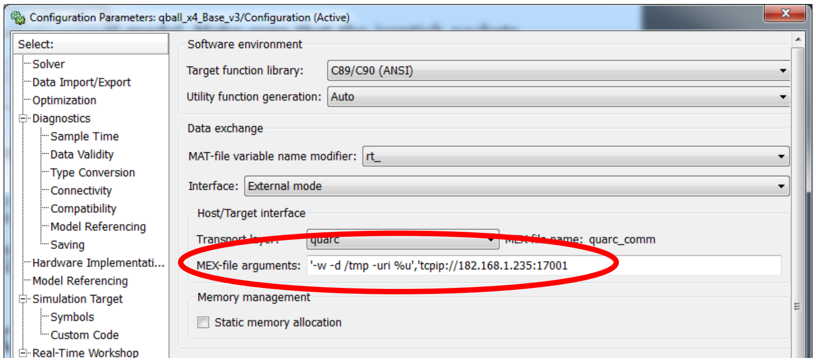


14. Connect the batteries and switch on the Qballs. There will be consistent “Beep-Beep” sound which will persist until the codes are written on the Qballs.
15. Click on wireless icon on the desktop taskbar and connect to the wireless network “GSAH.” (Refresh and wait a moment for the network to appear)
16. Back on the desktop and in Model ii), Double-click the block “Joystick from host” and followed by “Stream Client.” A dialog box as shown below should appear.



17. Check that the URI tcpip address should synchronize to the computer IP address. This can be verified by Start > “Under Search: type cmd” then press “Enter.” Type ipconfig and check the IPv4 Address. Enter the specific URI into the model in the URI entry as shown in the picture above. (Ensure that the correct port number as specified in the Host control model is consistently used.)

18. Go to Model (ii) and go to QUARC > Options on the menu list. The dialog as shown below appear



19. Check that the arguments ip address matches the QBall. There is a sticker on QBall rod to indicate its IP address.
20. Do the same for Model (iii), ensuring that the right port number is used.
21. Load the scene waypoints for the quads (e.g. “Scene 2.mat”). This contains the flight trajectory information for both systems.
22. Open the “console for all” panel to monitor the build process. Go to Quarc>console for all. A DOS screen will pop up if the qball is communicating with the host machine. If not, go back to step 15, and make sure that the connection is established before proceeding.
23. Click on “incremental build.” Wait for the compilation of the codes and transferring it to UAV A.
24. Once done, do the same for UAV B.
25. Click on “Connect to Target” and followed by “Start real-time code” to run Model (ii) and Model (iii). The “Beep-Beep” sound should stop.
26. Start the maneuver by moving the joystick’s throttle stick up to start the flight. For trajectory following, remember to start the flight soon after clicking on “start real time code.” Otherwise the quad may miss the front section of the flight and only perform the remaining flight.
27. UAV perform the flight plan.
28. When the flight id completed, move the joystick’s throttle stick down to land the UAV.
29. At any point if there is a need to stop the experiment, push the throttle down to land the UAV. Avoid pressing the stop button (in the matlab model) in the middle of a flight as it will cut command to the motor immediately.
30. Switch off the system and unplug the batteries.

THIS PAGE IS INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

7 July Review Committee. 2006. Report of the 7 July Review Committee: Greater London Authority.

Betts, J. T., and W. P. Huffman. 1997. *Sparse Optimal Control Software SOCS. Mathematics and Engineering Analysis,* . Technical Document: MEA-LR-085. Seattle: Boeing Information and Support Services, The Boeing Company.

Buede, Dennis M. 2009. *The Engineering Design of Systems*. Upper Saddle River, NJ: Wiley.

Castelli, Christopher. "DoD Report Predicts 'Robust Growth' for Unmanned Aircraft Industry," last modified September 26, 2012, accessed November, 2012, http://unmannedsystemsalert.com/Unmanned-Systems-General/Public-Content/dod-report-predicts-robust-growth-for-unmanned-aircraft-industry/menu-id-1004.html?S=LI.

Center for Disaster Philanthropy. "Hurricane Sandy Disaster Statistic." Center for Disaster Philanthropy, last modified January 15, 2013, accessed January, 2013, http://disasterphilanthropy.org/where/current-disasters/hurricane-sandy/hurricane-sandy-latest-news/.

Chua, C. N. 2012. Integration of Multiple UAVs for Collaborative ISR Mission in an Urban Environment. M.S. thesis, Systems Engineering, Naval Postgraduate School.

Cowling, I. D., J. F. Whidborne, and A. K. Cooke. 2006. "Proceedings of the International Control Conference: Optimal Trajectory Planning and LQR Control for a Quadrotor UAV." Glasgow, 30 August–11 September 2006.

DeBusk, Wesley M. 2009. Unmanned Aerial Vehicle Systems for Disaster Relief: Tornado Alley. Seattle: NASA.

Federal Emergency Management Agency. 2012. Technical Emergency Response Training for CBRNE Incidents. Vol. TERT.PM.112812.

Funato, Tatsuo. 2005. *Lessons Learned from Tokyo Subway Sarin Gas Attack*. Tokyo, Japan: Tokyo Metro.

Glenn, Russell W. 2002. *Urban Combat is Complex*. Santa Monica, CA: RAND Corporation.

Hargraves, C. R., and S. W. Paris. 1987. "Direct Trajectory Optimization using Nonlinear Programming and Collocation." *Journal of Guidance Control and Dynamics* 10 (4): 338–342.

Headquarters, Department of the Army. 2006. FM 3- 06 Urban Operations Headquarters Department of the Army.

International Crisis Group. 2006. Terrorism in Indonesia: Noordin's Networks: International Crisis Group.

Inuktun. "VGTV.," last modified 20 December 2012, accessed December, 2012, http://www.inuktunusa.com/crawler-vehicles/vgtv.html.

Jwa, Sangil, and Umit Ozguner. 2007. "Multi-UAV Sensing Over Urban Areas Via Layered Data Fusion." Madison, WI, USA, IEEE, 26–29 Aug. 2007.

Koo, T. J., and S. Sastry. 1999. "Proceedings of the 38th IEEE Conference Decision Control: Differential Flatness Based Full Authority Helicopter Design." Phoenix, AZ, 7–10 December.

Kossiakoff, Alexander, and William N. Sweet. 2002. *Systems Engineering: Principles and Practices*. Upper Saddle River, NJ: John Wiley & Sons.

Langford, Gary O. 2012. *Engineering Systems Integration: Theory, Metrics and Method*. Boca Raton: CRC Press.

Leng, Gerard, and Oleg Yakimenko. "Situational Awareness in Urban Areas." Situational Awareness in Urban Areas, .

Martin, Michael. 2012. "Global Versus Reactive Navigation for Joint UAV-UGV Missions in a Cluttered Environment." M. S. thesis, Mechanical Engineering, Naval Postgraduate School.

Milionis G. 2011. A Framework for Collborative Quadrotor-Ground Robot Missions. M. S. thesis, Naval Postgraduate School.

Mitsubushi Heavy Industries. "MHI Develops "MEISTeR " Disaster Recovery Support RobotWith 2 Arms Enabling Light-Duty Work Tasks." Press Information., last modified December 12 2012, accessed December 20, 2012, http://www.mhi.co.jp/en/news/story/1212121603.html.

Murphy, R., and S. Stover. 2006. "Gaps Analysis for Rescue Robots." AUVSI Unmanned Systems North America CRASAR-TR2005–12.

Murphy, R., S. Stover, and H. Choset. 2005. "Lessons Learned on the Uses of Unmanned Vehicles Fromthe 2004 Florida Hurricane Season." AUVSI Unmanned Systems North America.

Murphy, Robin R. 2004. "Trial by Fire: Activities of the Rescue Robots at the World Trade Center from 11–21 September 2001." *IEEE Robotics and Automation Magazine*.

NaturalPoint Corporation. 2011. *Tracking Tools User's Guide*. Vol. 2.3.3. Corvallis, OR: NaturalPoint Corporation.

NCTA, National Commission on Terrorist Attacks. 2004. The 9/11 Commission Report: Government Printing Office.

Nguyen, Hoa G., Robin Laird, Greg Kogut, John Andrews, Barbara Fletcher, Todd Webber, Rich Arrieta, and H. R. Everett. 2009. Land, Sea, and Air Unmanned Systems Research and Development at SPAWAR Systems Center Pacific. Orlando: SPAWAR.

NIST. 2005. Statement of Requirements for Urban Search and Rescue Robot: National Institute of Standards and Technology.

Paris, S. W., and C. R. Hargraves. 1996. *OTIS 3.0Manual.* Seattle: Boeing Space and Defense Group.

Quanser. 2011. *Quanser Qball-X4 User Manual*. Markham, Ontario: Quanser.

Quanser. 2009. *Quanser Unmanned Vehicles Systems Laboratory Information Package*: Markham, Ontario: Quanser.

Rice, Ian C. 2003. Urban Operations. M. S. thesis, Defense Analysis, Naval Postgraduate School.

Richards, A., and J. P. How. 2002. "In: Proceedings of the American Control Conference: Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming. ." Anchorage, AK, 8–10 May.

Robotic Trends News Source. "Networked Responders Robot Team Up to Map Disaster Sites.," last modified 9th Nov 2011, accessed Nov, 2012, http://www.roboticstrends.com/security_defense_robotics/article/networked_responder_robots_team_up_to_map_disaster_sites?&lang=en_us&output=json.

Ross, I. M. 2004. User's Manual for DIDO: A MATLAB Application Package for Solving Optimal Control Problems. Vol. TR 04–01.0 Tomlab Optimization Inc.

Saskatoon, Saskatchewan. "Dragonfly Introduces the X4-ES and X6-ES, an Emergency Services Exclusive UAV to Aid Police, Sheriff, and Fire and Rescue.." Dragonfly Innovations Inc., last modified September 29, 2012, accessed December, 2012, http://www.draganfly.com/news/2012/10/04/draganflyer-es-unmanned-helicopter-assists-fire-and-police-services/.

Savannah River National Laboratory. 2004. Urban Search and Rescue Technology Needs Identification of Needs.

Siciliano, Bruno, and Oussama Khatib. 2008. *Springer Handbook of Robotics*. Berlin: Springer.

Steimle, Eric T., Robin R. Murphy, Michael Lindemuth, and Michael L. Hall. 2009. “Unmanned Marine Vehicle use at Hurricanes Wilma and Ike.” *IEEE*, 26–29 Oct 2009.

Stopforth, Riaan, Glen Bright, and R. Harley. 2010. Communication and Artificial Intelligence Systems used for the CAESAR Robot. Mobile Robots Navigation., edited by Alejandra Barrera InTech.

The World Bank. “Urban Population (% of Total).” The World bank Data., last modified 2013, accessed January 4, 2013, http://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS/countries/1W?display=graph.

Trivedi, Bijal P. “Using Unmanned Subs to Probe the Deep.” National Geographic Today: Using Unmanned Subs to Probe the Deep., last modified August 16 2011, accessed November, 2012, http://news.nationalgeographic.com/news/2001/08/0816_NGTremus.html.

U.K. House of Commons. 2006. *Report of the Official Account of the Bombings in London on 7th July 2005*. London: The Stationary Office.

UNFPA. 2012. *State of World Population 2012*: UNFPA.

UNICEF. 2012. An Urban World UNICEF, http://www.unicef.org/sowc2012/urbanmap/.

Valenti, M., B. Bethke, G. Fiore, and J. P. How. 2006. “Proceedings of the AIAA Guidance, Navigation and Control Conference: Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery.” Keystone, Colorado, AIAA, .

VideoRay. “VideoRay ROV for Search & Recovery.,” last modified December 2012, accessed December, 2012, http://www.videoray.com/applications/law-enforcement/sar.html.

von Stryk, O. 2000. User’s Guide for DIRCOL (Version 2.1): A Direct Collocation Method for the Numerical Solution of Optimal Control Problems. Technische Universität Darmstadt.

Yakimenko, Oleg, Ian Cowling, James Whidborne, and Alastair Cooke. November 2010. “Direct Method Based Control System for an AutonomousQuadrotor.” *Journal of Intelligent and Robotic Systems* 60 (2): 285–316.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California